ESCOLA TÈCNICA SUPERIOR ENGINYERIA
UNIVERSITAT ROVIRA I VIRGILI
Departament d'Enginyeria Electrònica Elèctrica i Automàtica

Control Remoto Mediante Comunicaciones TCP/IP de Radioayudas Aeronauticas

AUTOR: Agustin Baubi Rovira .
DIRECTOR:Ernest Gil Dolcet .
DATA: Febrer 2002 .

Proyecto Final de Carrera:

CONTROL REMOTO MEDIANTE COMUNICACIÓN TCP/IP DE RADIOAYUDAS AERONAUTICAS

1.MEMORIA DESCRIPTIVA

- *Ingeniero Superior en Automática y Electrónica Industrial* — Agustín Baubí Rovira
- *Profesor Ponente* — Ernest Gil Dolcet

Febrero 2002

INDICE

Objetivos del proyecto 1
Titular y destinatario 2
Fundamentos sobre las Radioayudas Aeronáuticas 3
Configuración del Hardware 4
Configuración 1: Intranet Local 5
Configuración 2: Intranet con conexión a Internet a traves de un proxy 6
Configuración 3: Conexión a Internet a través de ISP 7
Fundamentos de TCP/IP y comunicación con sockets 8
Protocolos de red: Niveles 9
El Protocolo Internet (IP) 10
Protocolo de datagramas de usuario (UDP/IP) 11
Formato de direcciones IP: Orden de los bytes en red 12
Protocolo de control de transmisión (TCP/IP) 13
Fundamentos sobre el control MSWinsockControl 15
Fundamentos Sobre Bases de Datos 19
Fundamentos del Puerto Paralelo 25
Breve descripción del puerto paralelo: 26
Implementación de la clase CPuertoParalelo 28
Funcionamiento de la aplicación 29
Dialogo principal. Aplicación servidor 30
Diálogo Base de Datos 32
Diálogo Propiedades 33
Diálogo Cliente 34
Solución al problema de la IP dinámica 36
Conclusiones 39
Posibles líneas de continuación del Proyecto 40

Objetivos del proyecto

El presente proyecto pretende realizar una aplicación informática para ejecutar el control y monitorización remota de unos equipos electrónicos aeronáuticos desde un PC conectado en una red Ethernet.

Para ello el software realizado en Microsoft Visual C++ debe ser capaz de controlar el puerto paralelo del PC, que se encuentra en la instalación de la radioayuda y de comunicarse con otra aplicación situada en el lugar donde se centraliza el control y monitorización de todas las instalaciones.

Mediante el puerto paralelo del PC de la instalación dispondremos de ocho salidas y cuatro entradas digitales con las cuales podremos generar las ordenes de control pertinentes y monitorizar el estado de la instalación.

El programa realizado actuará como programa servidor, a la escucha de que otro ordenador cliente conectado a la red intente establecer una conexión. El otro ordenador cliente tendrá la misma aplicación instalada pero funcionando como programa cliente.

Una vez realizada la conexión se podrá modificar el estado de las salidas y recibir el estado de las entradas del puerto del ordenador servidor remoto así como adaptar el diálogo cliente al tipo de instalación que se ha conectado, indicando la función de cada entrada y salida mediante una leyenda actualizada. También realiza un registro sobre una base de datos de todos los eventos que producen las entradas y salidas.

El tipo de red de ordenadores que se utiliza es cualquier tipo de red Ethernet basada en el protocolo TCP/IP. Así pues el ordenador servidor y cliente pueden realizar una conexión mediante una red WAN (*Wide Area NetWork*) como Internet o una red LAN (*Local Area NetWork*) tipo una intranet privada de una empresa.

La funcionalidad de esta aplicación como un Sistema Informático en Tiempo Real estará basada en la capacidad de asegurar un tiempo mínimo de comunicación entre los dos ordenadores. Difícilmente se puede establecer tiempos límites en aplicaciones conectadas en Internet pero en redes LAN basadas en red ATM (*Asyncrhonous Transfer Mode*) si que se puede limitar el tiempo de la comunicación y por tanto asegurarnos de que este tiempo es inferior al tiempo que necesita el sistema que queremos controlar.

La utilidad de este proyecto se centra en ejercer como una potencial aplicación industrial inmediata en el sector aeronáutico pero también en ofrecer una solución válida para otros tipos de sectores y ofreciendo también una base para estudios posteriores, tanto para la utilización en otras aplicaciones informáticas como en la ampliación de este proyecto con opciones más sofisticadas.

Titular y destinatario

El posible destinatario de este proyecto sería AENA (Aeropuertos Españoles y Navegación Aérea) empresa en la cual estoy trabajando desde hace cuatro años en el Aeropuerto de Reus.

Trabajando en las dependencias de Navegación Aérea se observa la necesidad de monitorizar y controlar remotamente las diferentes instalaciones electrónicas que se utilizan para ofrecer servicio de navegación a las aeronaves. Para ello se utilizan sistemas de comunicación punto a punto mediante líneas telefónicas o radio enlaces que conectan las radioayudas con monitores centralizados en la Sala de Equipos de las Torres o Centros de Control.

Con la ayuda del software realizado en este proyecto se podría utilizar la red informática interna que dispone la empresa y desde un ordenador central en la Sala de Equipos controlar las diferentes radioayudas instaladas en el Aeropuerto, y es más, se puede tener una comunicación con la instalación desde cualquier punto de la empresa con cualquier ordenador conectado a la misma red y con este software instalado.

También añadiríamos la función de almacenamiento de un histórico, sobre una base de datos, con todos los eventos que se producen en cada instalación. Indispensable para poder realizar una buena acción de mantenimiento sobre los equipos.

De esta manera instalando en todas las instalaciones de radioayudas un enlace a la red informática de la empresa se podría centralizar en un lugar la monitorización de todas las radioayudas que dispone la empresa.

Otro posible destinatario de este proyecto es la propia Universidad URV, en concreto el Departamento de Enginyeria Elèctrica, Electrònica i Automàtica, donde futuros alumnos e investigadores en este tema puedan continuar con la mejora de este proyecto o utilizar el código generado para incorporarlo en otros proyectos para ofrecer servicios de comunicación entre aplicaciones en redes informáticas basadas en el protocolo TCP/IP.

Fundamentos sobre las Radioayudas Aeronáuticas

Las Radioayudas Aeronáuticas son diferentes instalaciones equipadas con equipos electrónicos de radiofrecuencia que proporcionan a las aeronaves diferentes servicios de ayuda a la navegación aérea como puede ser la generación de las diferentes aerovías para la navegación (VOR), distancias de la aeronave a un punto de destino (DME), orientación (NDB), ayuda al aterrizaje (ILS), localización (RADAR), etc

Todas estas instalaciones están dotadas de equipos redundantes para asegurar un funcionamiento continuo en todo momento.

Las principales señales generadas por los equipos electrónicos son auto comprobadas por unos equipos monitores internos que tienen todas estas instalaciones. En caso de que este monitor detecte una señal fuera de sus límites de tolerancia conmuta automáticamente al equipo de reserva. Este sistema interno de monitorización puede ser deshabilitado (bypass) para realizar diferentes actuaciones de mantenimiento, pero mientras los equipos están ofreciendo servicio a las aeronaves los monitores deben estar siempre habilitados.

Los equipos pueden estar conectados a la antena, emitiendo las señales útiles para la navegación, o conectados a una carga, para mantener los sistemas electrónicos en un estado de funcionamiento pero sin producir ninguna radiación.

Las principales señales del equipo que se deben incluir en un control remoto de control (salidas del puerto) y monitorización (entradas del puerto) son:

Entradas	Salidas
Entrada1: Equipo ON/OFF	Salida1: Equipo Principal ON en Antena
Entrada2: Equipo en antena Principal / secundario	Salida2: Equipo Principal ON en Carga
Entrada3: Monitores Bypass	Salida3: Equipo Principal OFF
Entrada4: Alarma de Mantenimiento	Salida4: Equipo Secundario ON en Antena
	Salida5: Equipo Secundario ON en Carga
	Salida6: Equipo Secundario OFF
	Salida7: Monitores Bypass
	Salida8: Reset alarma de Mantenimiento

El software realizado en este proyecto controla el puerto paralelo de un PC y ofrece ocho salidas y cuatro entradas digitales directas para controlar las señales anteriores. En caso de necesitar más señales que las ofrecidas se puede utilizar el método de codificación de las señales, así podríamos disponer de $2^8 = 256$ salidas y $2^4 = 16$ entradas digitales codificadas.

También cabe la posibilidad de instalar más de un PC en una determinada instalación.

Las señales del puerto paralelo deben cablearse a los equipos de las radioayudas mediante las adaptaciones necesarias en cada caso para capturar la información necesaria.

Configuración del Hardware

El software realizado (ProyectoBD.exe) se debe instalar en un ordenador PC compatible con el sistema operativo Windows 95 o superior.

Para conseguir controlar remotamente las instalaciones se pueden utilizar diferentes configuraciones del hardware.

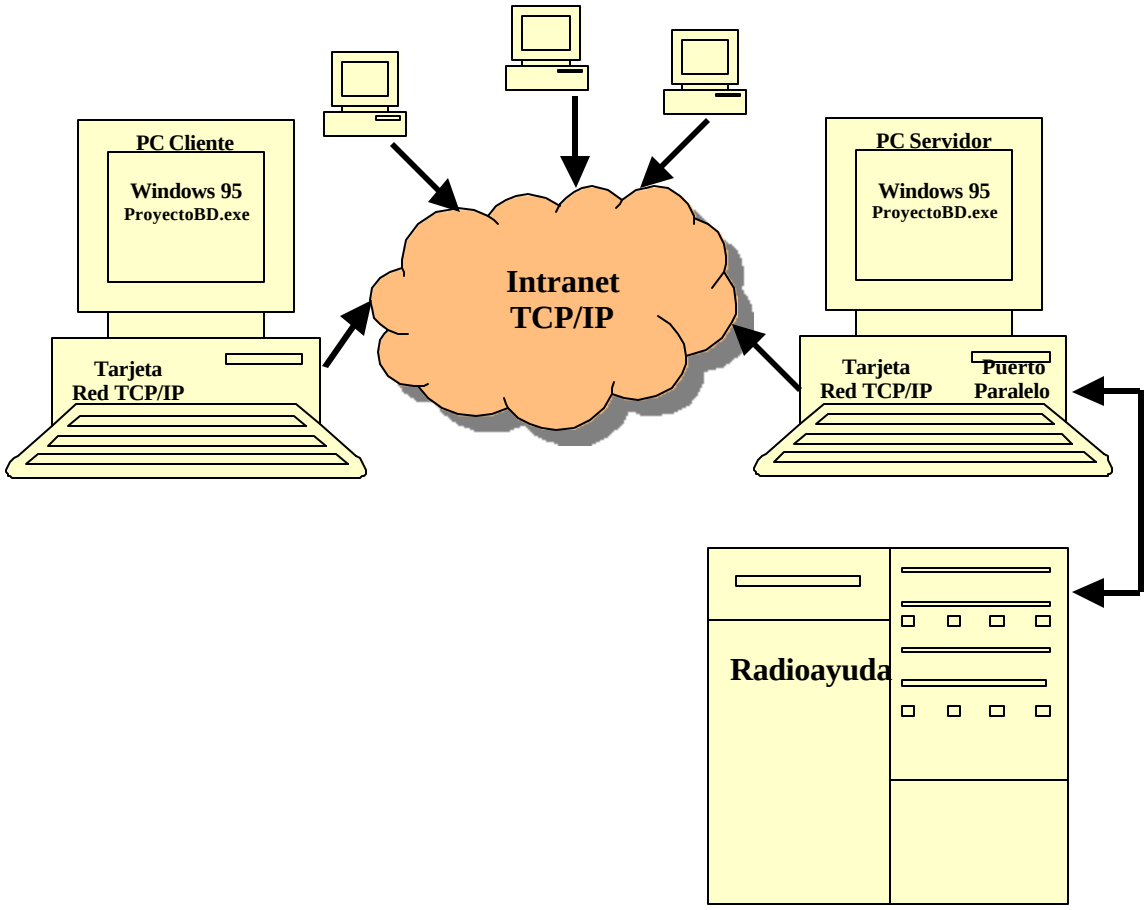
Según el caso, el hardware para conectarse a la red TCP/IP es diferente.

1. *Intranet Local*
   Podemos enlazar con la red TCP/IP directamente con una tarjeta de red a nuestra Intranet Local

2. *Intranet con conexión a Internet a traves de un proxy*
   Enlazamos con la red TCP/IP con la tarjeta de red a nuestra Intarnet Local y con la ayuda de un servidor proxy a Internet

3. *Conexión a Internet a través de ISP*
   Conectarnos telefonicamente mediante un módem a un Proveedor de Servicios de Internet (ISP) que enlazará nuestro PC a la red Internet.

Configuración 1: Intranet Local

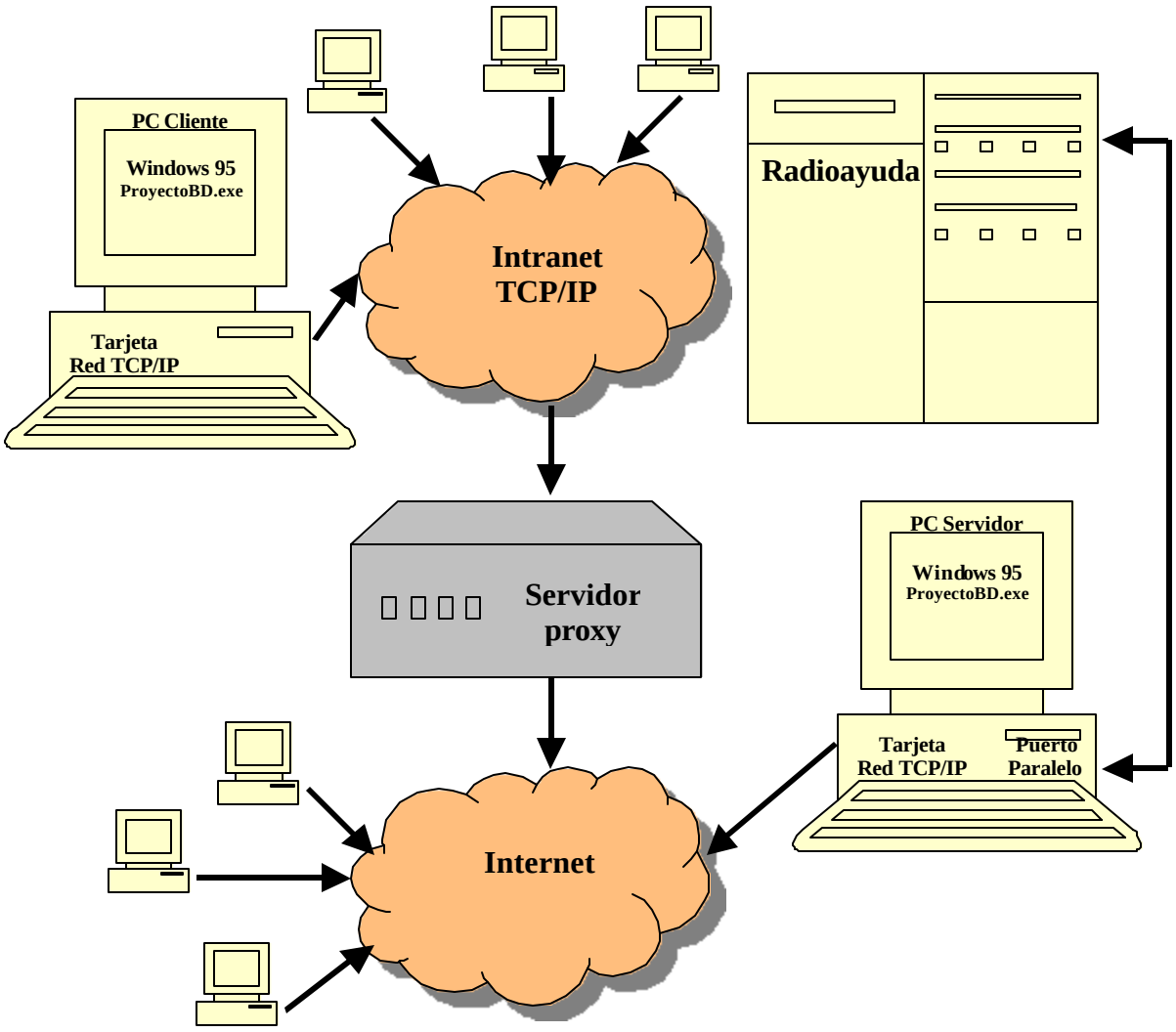


En este tipo de configuración el ordenador cliente y servidor están conectados a una Intranet local a través de una tarjeta adaptadora de red.

El administrador de la red se encarga de proporcionar una dirección IP estática y privada a cada PC conectado a la Intranet. (En puntos posteriores de esta memoria se explica el significado de la dirección IP y de los diferentes tipos que existen; Pág. 8, Pág. 36).

Sabiendo que dirección IP es la que tiene el PC servidor y fijando arbitrariamente el puerto que vamos a utilizar podrá establecerse la comunicación entre los dos PC.

Configuración 2: Intranet con conexión a Internet a través de un proxy



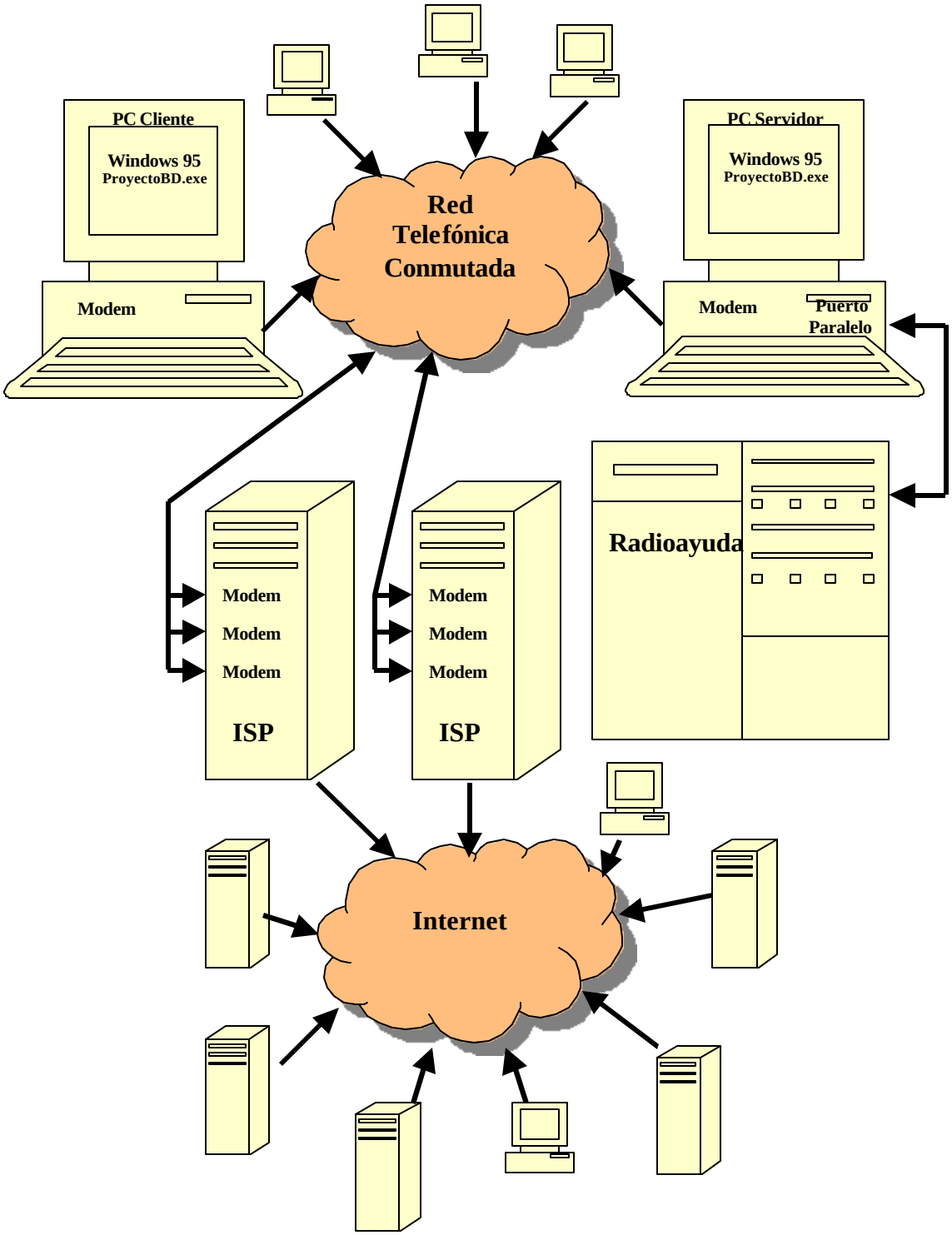
Esta configuración es parecida a la anterior pero en este caso el ordenador servidor esta conectado a la red mundial Internet.

El PC servidor dispone de una dirección IP pública y estática, siempre es la misma y no hay otra igual en toda Internet.

Nuestro PC cliente dispone de una IP estática, la de la Intranet Local, que no es pública, para salir a Internet utiliza un servidor proxy, que se encarga de dar acceso a Internet a los PC de la Intranet. El proxy es la única máquina de la Intranet que dispone de IP pública.

Sabiendo que dirección IP es la que tiene el PC servidor y fijando arbitrariamente el puerto que vamos a utilizar podrá establecerse la comunicación entre los dos PC.

Configuración 3: Conexión a Internet a través de ISP



En este caso los ordenadores cliente y servidor se conectan a Internet a traves de un proveedor de servicios de Internet (ISP, como Tinet o Terra). Para ello utilizan un modem que se conecta por la red telefonica con el ISP, que proporcina las direcciones IP públicas y dinámicas. (Solución al problema de la IP dinámica pag. 36)

Fundamentos de TCP/IP y comunicación con sockets

Hoy en día todo el mundo conoce Internet, la red mundial basada en el protocolo TCP/IP, desde donde se tiene acceso a un ilimitada fuente de información.

Con el boom de Internet parece que todas las aplicaciones basadas en cliente-servidor solo se utilizan en Internet pero no hay que pasar por alto que es posible ejecutar exactamente el mismo software en una intranet local.

- intranet.-- Red privada que utiliza los protocolos TCP/IP. Puede tener salida a Internet o no. En el caso de tener salida a Internet, el direccionamiento IP permite que los hosts con direcciones IP privadas puedan salir a Internet pero impide el acceso a los hosts internos desde Internet. Dentro de una intranet se pueden configurar todos los servicios típicos de Internet (web, correo, mensajería instantánea, etc.) mediante la instalación de los correspondientes servidores. La idea es que las intranets son como "Internets" en miniatura o lo que es lo mismo, Internet es una intranet pública gigantesca.

- Extranet.-- Unión de dos o más intranets. Esta unión puede realizarse mediante líneas dedicadas (RDSI, X.25, frame relay, punto a punto, etc.) o a través de Internet.

- Internet.-- La mayor red pública de redes TCP/IP.

Las intranet suelen implementarse como redes LAN internas de una empresa sobre las que se montan aplicaciones distribuidas. Los usuarios disponen de PC con aplicaciones cliente, mientras que los PC servidores suministran o ejecutan procesos de datos en respuesta a las solicitudes de los clientes. Para salvaguardar la información y los datos de la empresa se pueden separar completamente la intranet de Internet conectándolas a través de cortafuegos o proxy's

Mediante el control winsock podemos crear aplicaciones distribuidas aprovechando todo el potencial de las redes TCP/IP (Internet e intranet)

Cualquiera que quiera dedicarse al desarrollo de software para Internet no puede pasar por alto el control winsock basado en el concepto de socket (conector)

El socket permite a una aplicación establecer una conexión con cualquier punto de la red, así como enviar y recibir datos, principales funciones de cualquier aplicación de Internet.

Para entender correctamente que es un socket hace falta ciertos conocimientos de los protocolos que se utilizan en Internet.

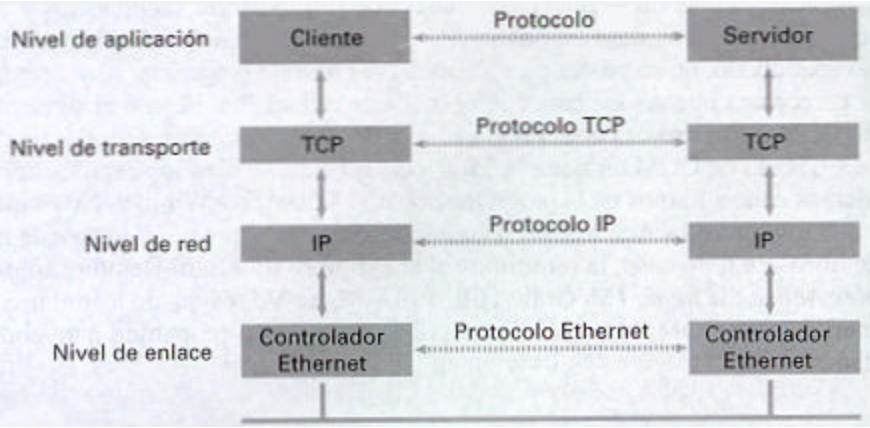
Protocolos de red: Niveles

Todas las redes descomponen los protocolos de transmisión en niveles, al conjunto de niveles se le llama frecuentemente torre.

La aplicaciones se comunican con el nivel superior de la torre mientras que los niveles inferiores se comunican por la red.

Cada nivel se conecta desde el punto de vista lógico con el mismo nivel en el otro extremo del canal de comunicaciones (cliente-servidor)

Una red de área local (LAN) basada en TCP/IP esta formada por una torre con los siguiente niveles



Torre de protocolos de una LAN ejecutando TCP/IP

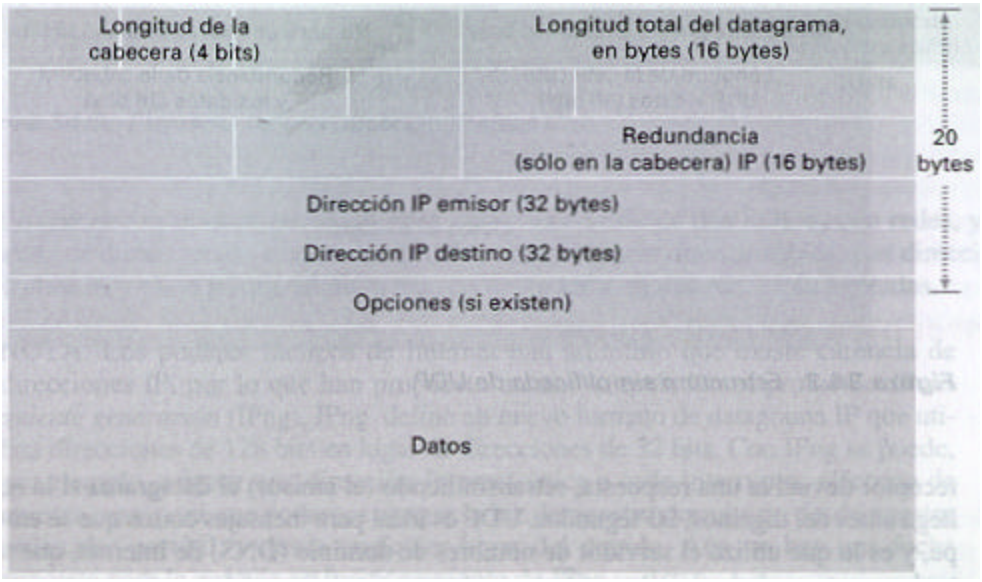
El programa servidor escucha continuamente por un extremo del canal, mientras que el programa cliente se conecta periódicamente con el programa servidor para intercambiar datos.

Ejemplo: El servidor puede ser un servidor de páginas web basado en http y el cliente un programa Navegador.

El Protocolo Internet (IP)

El nivel de protocolo Internet (IP) es por donde se empieza a estudiar TCP/IP.

El protocolo IP define paquetes, datagramas, que constituyen las unidades fundamentales en una comunicación en Internet. Estos paquetes, inferiores a 1Kbyte, van rebotando por todo el mundo cuando se esta ejecutando una aplicación de Internet (se abre una página web, se carga un fichero vía FTP, se envía un correo electrónico, etc.)


Longitud de la cabecera (4 bits)
Longitud total del datagrama, en bytes (16 bytes)
Redundancia (sólo en la cabecera) IP (16 bytes)
20 bytes
Dirección IP emisor (32 bytes)
Dirección IP destino (32 bytes)
Opciones (si existen)
Datos

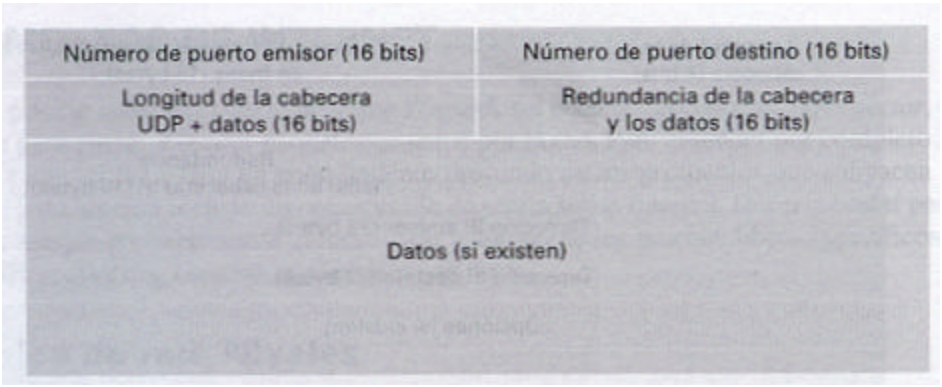
Estructura simplificada de un datagrama IP

El datagrama IP contiene direcciones de 32 bits tanto para la computadora origen como para la destino. Estas direcciones IP identifican unívocamente a las computadoras en Internet y son utilizadas por los encaminadores (Routers, computadoras especializadas que funcionan como conmutadores telefónicos) para dirigir los datagramas individuales a sus destinos. Los encaminadores no se preocupan por los que hay dentro del datagrama, solo están interesados en la dirección de destino del datagrama y en su longitud total. Su función es reenviar el datagrama lo más rápidamente posible.

El nivel IP no le dice al programa emisor si el datagrama llego con éxito a su destino. Ésa es la función del nivel superior de la torre. El programa receptor sólo puede comprobar la redundancia para determinar si la cabecera del datagrama IP se ha corrompido.

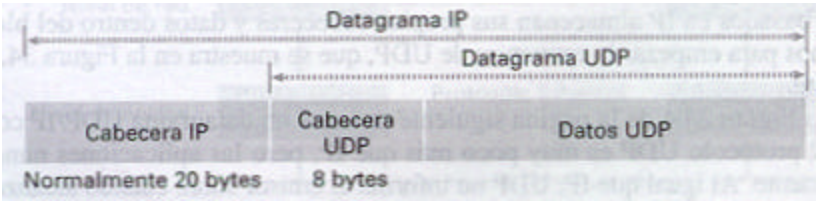
Protocolo de datagramas de usuario (UDP/IP)

El protocolo TCP/IP debería llamarse en realidad TCP/UDP/IP, ya que incluye el protocolo de datagramas de usuario (UDP), que es alternativa a TCP. Todos los protocolos de transporte basados en IP almacenan sus propias cabeceras y datos dentro del bloque de datos IP.


Número de puerto emisor (16 bits)
Número de puerto destino (16 bits)
Longitud de la cabecera UDP + datos (16 bits)
Redundancia de la cabecera y los datos (16 bits)
Datos (si existen)

Estructura simplificada de UDP

La figura anterior muestra como la cabecera UDP incluye algo de información adicional, en concreto, los números de puerto origen y destino. Estos números de 16 bits se utilizan en los programas de aplicación en ambos extremos. Por ejemplo, un programa cliente podría enviar un datagrama al puerto 1700 del servidor. El programa servidor permanece a la escucha de cualquier datagrama que incluya 1700 como puerto de destino y, cuando encuentra uno, puede responder enviando otro datagrama al cliente, que escuche aquellos datagramas que incluyan 1701 como número de puerto de destino.

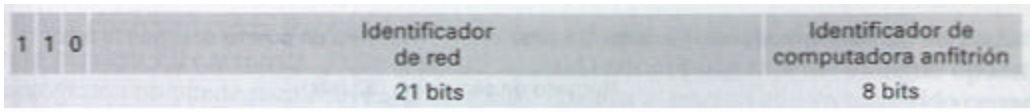


Datagrama UDP/IP completo. Relación entre el datagrama IP y el datagrama UDP

El protocolo UDP es muy poco más que IP, pero las aplicaciones nunca utilizan IP directamente. Al igual que IP, UDP no informa al emisor sobre cuándo alcanza el datagrama su destino. Eso es parte del trabajo de la aplicación. Por ejemplo, el emisor podría esperar a que el receptor emitiera una respuesta, retransmitiendo el emisor el datagrama si la respuesta no llega antes de, digamos , 20 segundos. UDP es ideal para mensajes cortos que se envían de golpe, transmisión de audio y video en tiempo real para los que la pérdida de un datagrama o el desorden entre ellos no constituye un excesivo problema.

Formato de direcciones IP: Orden de los bytes en red

Ya sabemos que las direcciones IP son de 32 bits. Esto significa que pueden coexistir en Internet 2^{32} (más de cuatro billones) máquinas con dirección unívoca, aunque esto no es del todo cierto. Parte de la dirección identifica la LAN en la que reside la computadora anfitrión y parte de ella identifica a la computadora anfitrión (host) dentro de la red. La mayoría de direcciones IP son direcciones clase C, cuyo formato es el que se muestra a continuación.



Esquema de una dirección IP clase C

Así pues, pueden existir poco más de dos millones de redes, y que cada una de dichas redes puede tener 2^8 (256) computadoras direccionables. Las direcciones IP de la clase A y clase B, que admiten más computadoras en una red están agotadas.

Los responsables de Internet han admitido que existe carencia de direcciones IP, por lo que han propuesto un nuevo estándar, el protocolo IP siguiente generación (IPng) que define un nuevo formato de datagrama IP que utiliza direcciones de 128 bits en lugar de 32 bits. Con IPng se puede, por ejemplo, asignar una dirección Internet única a cada interruptor eléctrico de nuestra casa, para que podamos apagar la luz de nuestro dormitorio desde nuestro ordenador portátil, y desde cualquier lugar del mundo. Aún no hay una fecha concreta para la entrada en funcionamiento de IPng.

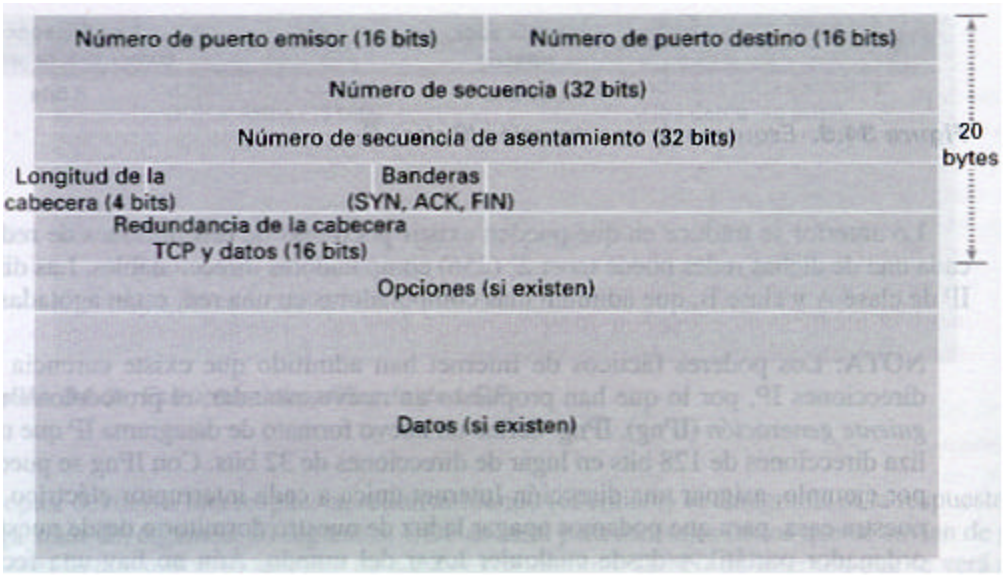
Por convención las direcciones IP se escriben en formato punto decimal. A cada una de las cuatro partes de la dirección le corresponde un valor byte individual. Por ejemplo, una dirección IP de clase C sería 194.128.198.201

Cuando navegamos por la Web no utilizamos direcciones IP. En su lugar utilizamos nombres más humanizados, del tipo www.etse.urv.es. Una parte significativa de los recursos de Internet se consumen en la traducción de nombres de maquina (www.urv.es) a direcciones IP que TCP/IP pueda utilizar. Esta traducción la lleva a cabo un conjunto de servidores de nombres (DNS, servidores de dominio). La totalidad del espacio de nombres de Internet se organiza en dominios, cuyo comienzo se sitúa en el dominio raíz, que carece de nombre. Por debajo de la raíz hay una serie de dominios de alto nivel, del tipo com, edu, gov y org.

Protocolo de control de transmisión (TCP/IP)

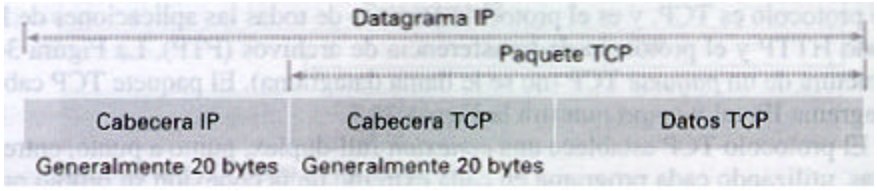
Vistas las limitaciones de UDP, lo que realmente necesitamos es un protocolo que permita transmitir grandes bloques de datos sin errores. El programa receptor tiene que ser capaz de reensamblar los bytes en el mismo orden y secuencia en que fueron transmitidos, incluso aunque los datagramas anteriores hayan llegado en orden incorrecto.

Ese protocolo es TCP, y es el protocolo favorito de todas las aplicaciones de Internet


Número de puerto emisor (16 bits)
Número de puerto destino (16 bits)
Número de secuencia (32 bits)
Número de secuencia de asentamiento (32 bits)
Longitud de la cabecera (4 bits)
Banderas (SYN, ACK, FIN)
Redundancia de la cabecera TCP y datos (16 bits)
Opciones (si existen)
Datos (si existen)
20 bytes

Estructura básica de un paquete TCP/IP

El paquete TCP cabe dentro de un datagrama IP



Relación entre un datagrama IP y un segmento TCP

El protocolo TCP establece una conexión full-duplex, punto a punto, entre dos computadoras, utilizando cada programa en cada extremo de la conexión su propio puerto. La combinación de una dirección IP con un número de puerto recibe el nombre de conector (socket) La conexión se establece al inicio mediante un protocolo de tres fases. El programa que inicia la conexión envía un paquete con la bandera SYN activada. El programa que responde devuelve un paquete con las banderas SYN y ACK activadas y, por último, el programa que inicia la transmisión envía un paquete con la bandera ACK activada.

Tras el establecimiento de la conexión, cada programa puede enviar un flujo de bytes al otro programa. TCP utiliza los campos de numero de secuencia junto con las banderas ACK para gestionar ese flujo de bytes. El programa que envía la transmisión no espera a que cada paquete sea asentido, sino que envía un número de paquetes consecutivamente; después espera hasta el primer asentamiento. Si el programa receptor tiene datos para enviar al otro programa, puede montar el asentamiento sobre el paquete de datos y transmitirlo todo junto.

La secuencia de números del programa transmisor no es de índice de paquete, sino más bien de índices de flujo de bytes. El programa receptor devuelve los números de secuencia (en el campo numérico del asentamiento) al programa emisor, asegurando con ello que todos los bytes se reciben y ensamblan de manera adecuada. El programa emisor re-envía los paquetes que no hayan sido asentidos.

Cada programa cierra la conexión TCP en su extremo mediante el envío de un paquete con la bandera FIN activada, y que debe ser asentida por el programa del extremo opuesto. Un programa no puede recibir bytes provenientes de una conexión que ya haya sido cerrada por el programa del otro extremo.

En Control Winsock oculta la mayor parte de los detalles del protocolo TCP/IP por los que no tenemos que preocuparnos de las banderas ACK ni de los números de secuencia. Nuestro programa llama a una función para que transmita un conjunto de datos, y Winsock es quien se preocupa de dividir el bloque en segmentos que, a continuación, incrusta en datagramas IP. Winsock también se ocupa de distribuir los bytes en el extremos receptor.

Fundamentos sobre el control MSWinsockControl

El MSWinsockControl incluido con Visual C++ es una herramienta muy poderosa para la creación de aplicaciones basadas en sockets.

Con MSWinsockControl se pueden implementar los diferentes protocolos utilizados en Internet e inclusive protocolos propios.

Los principales protocolos de Internet basados en TCP/IP son:

HTTP	Hypertext Transfer Protocol (páginas Web)	puerto 80
FTP	File Transfer Protocol (transferencia de ficheros)	puerto 21
SMTP	Simple Mail Transfer Protocol (envío de email)	puerto 25
POP3	Post Office Protocol (recepción email)	puerto 110
NNTP	Network News Transfer Protocol (noticias)	puerto 119

Para establecer la comunicación se procede normalmente de la siguiente manera:

Lo primero es que el programa servidor esté escuchando en el puerto correspondiente (servidor de paginas web, etc), después, algún programa cliente (navegador web, etc) se conecta al servidor tras recibir la dirección IP del mismo desde un servidor de nombres y utilizando el número de puerto el cliente establece la conexión.

Otros protocolos como el FTP establecen dos conexiones distintas entre el cliente y el servidor, una para datos y otra para comandos.

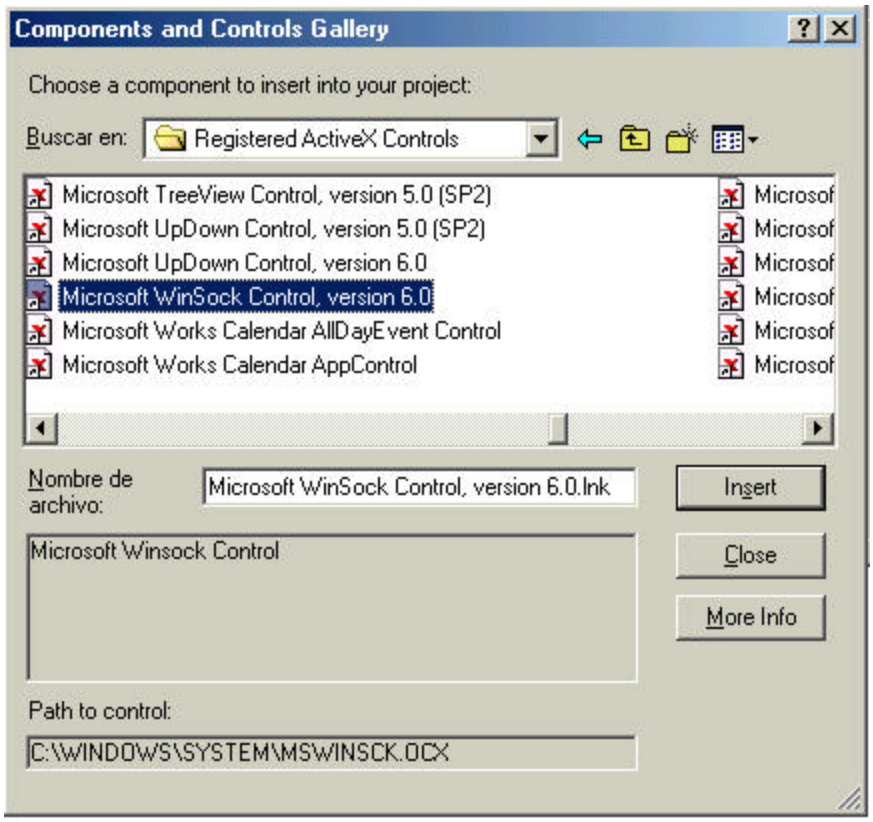
Las aplicaciones que desarrollemos pueden utilizar tantos sockets como sean necesarios para realizar la tarea encomendada de la manera más eficiente pero siempre teniendo en cuenta que cada socket trabaja sobre un único puerto.

En nuestras aplicaciones utilizaremos un puerto elegido al azar pero para evitar confusiones intentaremos evitar en lo posible puerto utilizados por otras aplicaciones.

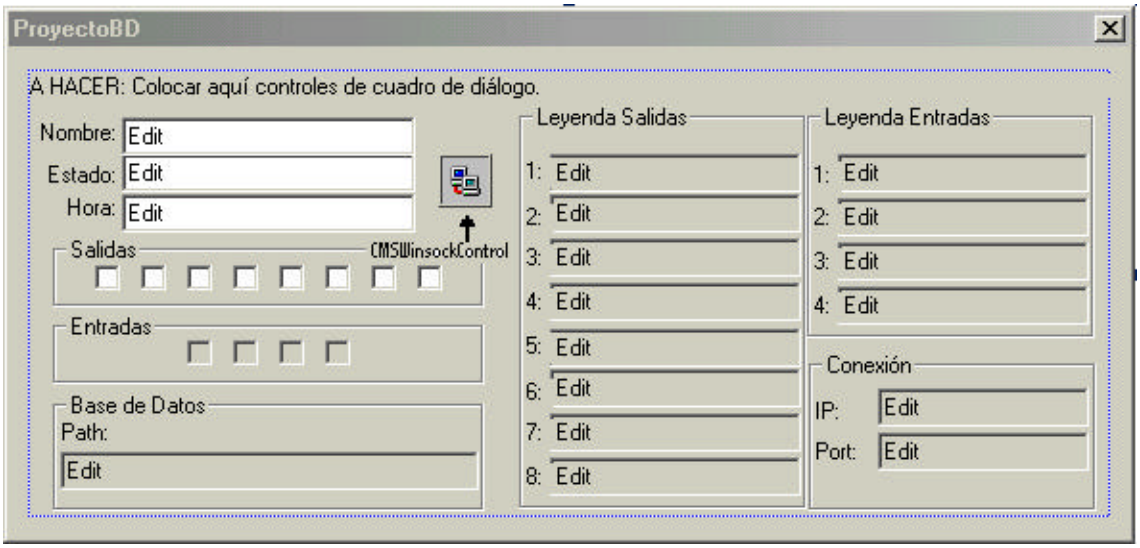
Para introducir el control MSWinsockControl en una aplicación Visual C++ realizaremos los siguientes pasos

-Insertar Control Winsock

Menu Project->Add to Project -> Components and Controls-> Registered ActiveX Controls -> Microsoft Winsock Control

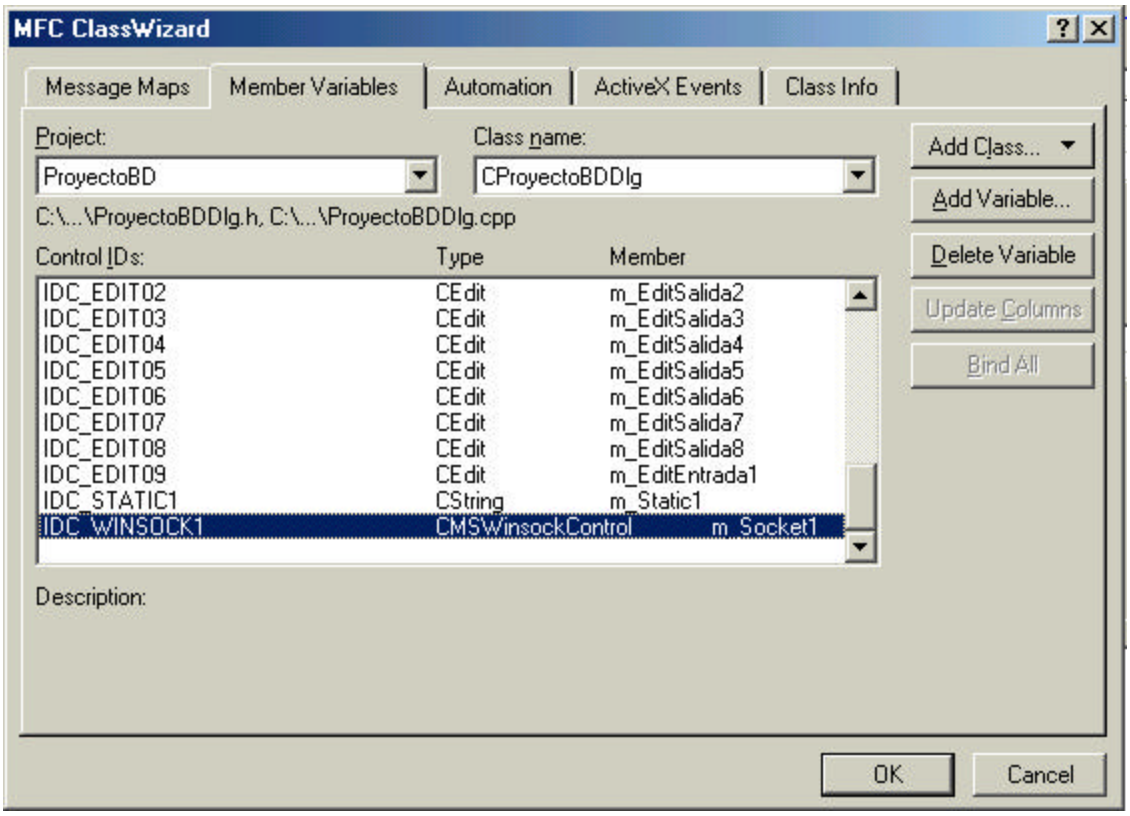


-Insertar control CMSWinsockControl en el dialogo del proyecto



-Clear variable *m_Socket1;* (classwizard, member variables)

IDC_WINSOCK1 *m_Socket1* *Control*



la variable creada m_Socket1 es una instancia de la clase CMSWinsockControl y por la tanto hereda todas las propiedades, métodos y eventos de dicha clase.

Propiedades	Métodos	Eventos
Bytesreceived	Accept	Close
Localhostname	Close	Connectionrequest
Localip	Connect	Dataarrival
Localport	Getdata	Error
Name	Listen	Sendcomplete
Parent	Senddata	Sendprogress
Protocol		
Remotehost		
Remotehostip		
Remoteport		
Sockethandle		
State		
Servicename		

Para facilitar la programación hemos creado unas funciones propias, incluidas en el fichero *ABRWinsock.h* que añadimos a la clase CMSWinsockControl que facilitan la programación del socket en cualquier aplicación

Principales funciones creadas:

CString Dato;

Variable donde almacenamos los datos que se envían y reciben a través del socket

ABRConectarSck();

Función para conectar con el socket servidor, en el interior de esta función es donde se establecen la dirección IP del servidor y el número de puerto del socket

ABREscucharSck();

Función para poner al servidor en estado de escucha de algún cliente que se quiera conectar

ABROnConnectionRequestWinsock1(long requestID);

Función que detecta que algún cliente esta intentando establecer la conexión con el socket y acepta la comunicación

ABROnDataArrivalWinsock1(long bytesTotal);

Función que detecta la llegada de datos y las almacena en la variable dato

ABREnviarSck(CString Datos);

Función para enviar datos a través del socket

ABRCerrarSck();

Función que cierra la conexión realizada entre el cliente y el servidor

ABRConvertirEstado(int Estado);

Función que devuelve una cadena de texto con el estado en que se encuentra el socket

Fundamentos Sobre Bases de Datos

La necesidad de registrar en un histórico los diferentes eventos que se producen en las instalaciones a controlar ayuda a mejorar las tareas de mantenimiento tanto correctivo como preventivo necesario en las instalaciones de Navegación Aérea.

Se podría haber realizado el histórico de los eventos mediante un acceso a ficheros en disco (serialización) pero se ha escogido utilizar una base de datos debido a la posibilidad de utilizar toda la potencia que disponen las aplicaciones de gestión de bases de datos para sacar todo el partido a los datos almacenados y la posibilidad de utilizar el estándar ODBC que facilita utilizar diferentes bases de datos.

Hoy en día los programadores de aplicaciones para Windows pueden escoger entre un amplio repertorio de sistemas de gestión de bases de datos entre los que podríamos destacar Inprise Paradox, Microsoft Access, Microsoft FoxPro y PowerSoft PowerBuilder. La mayor parte de estos productos son capaces de acceder tanto a datos locales como a datos remotos que residan en una computadora central. en este último caso, se precisa además el software servidor de bases de datos, como por ejemplo ORACLE o Microsoft SQL Server.

Visual C++ incluye herramientas y clases MFC de todos los componentes precisos para escribir aplicaciones de base de datos. Esto incluye dos sistemas distintos de acceso a bases de datos (ODBC, Open Database Connectivity), Objeto de acceso a datos (DAO, Data Access Objects) y OLE DB basada en el modelo de objetos de componentes (COM).

El estándar ODBC, que será el utilizado, consiste en un conjunto extensible de bibliotecas de enlace dinámico (DLL) que proporcionan un interfaz de programación de bases de datos normalizada del lenguaje estructurado de consultas (SQL, Structured Query Language). mediante ODBC y SQL es posible escribir código de acceso a bases de datos independiente del producto de bases de datos.

SQL es un lenguaje estándar de acceso a base de datos, con gramática propia. En el mundo SQL, una base de datos es un conjunto de tablas, compuestas por filas y columnas. La mayor parte de productos de gestión de base de datos soportan SQL, lenguaje conocido por muchos programadores.

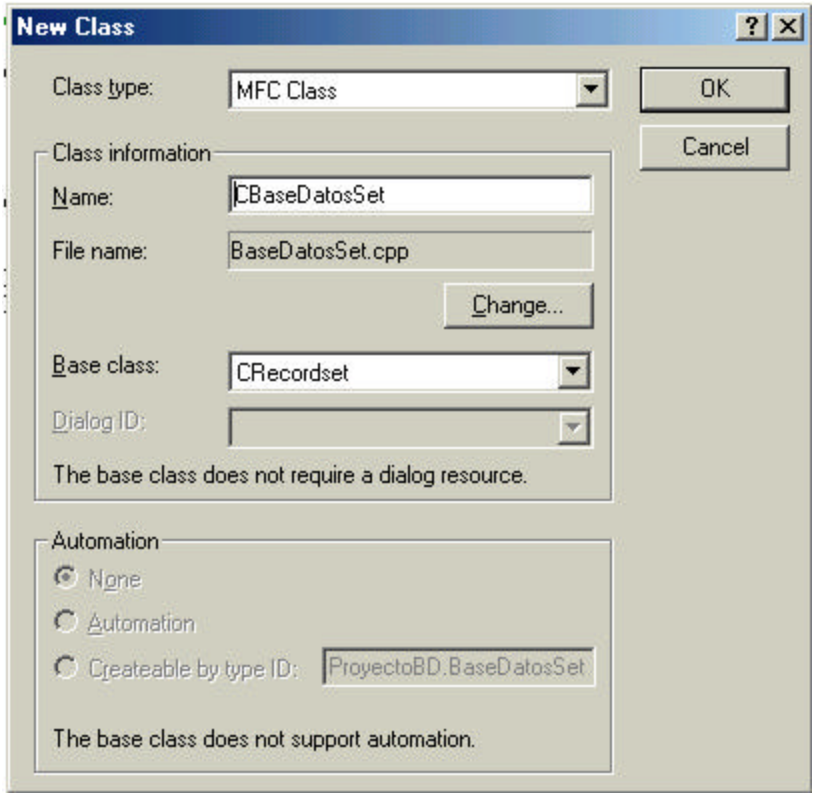
Un único programa en Visual C++ compilado puede acceder a cualquier producto de gestión de base de datos que disponga de un controlador ODBC. Visual C++ incorpora controladores de 32 bits para archivos DBF, bases de datos Microsoft Access MDB, archivos XLS de Microsoft Excel, archivos de Microsoft FoxPro, archivos de texto ASCII y bases de datos de Microsoft SQL Server. Otras empresas incluyendo Oracle, Informix, etc proporcionan controladores ODBC. Así pues si una aplicación MFC de Visual C++ se desarrolla con un controlador dBASE/Xbase, el mismo programa se puede ejecutar con un controlador de base de datos Access, no es necesario recompilarlo, el programa simplemente carga una DLL distinta.

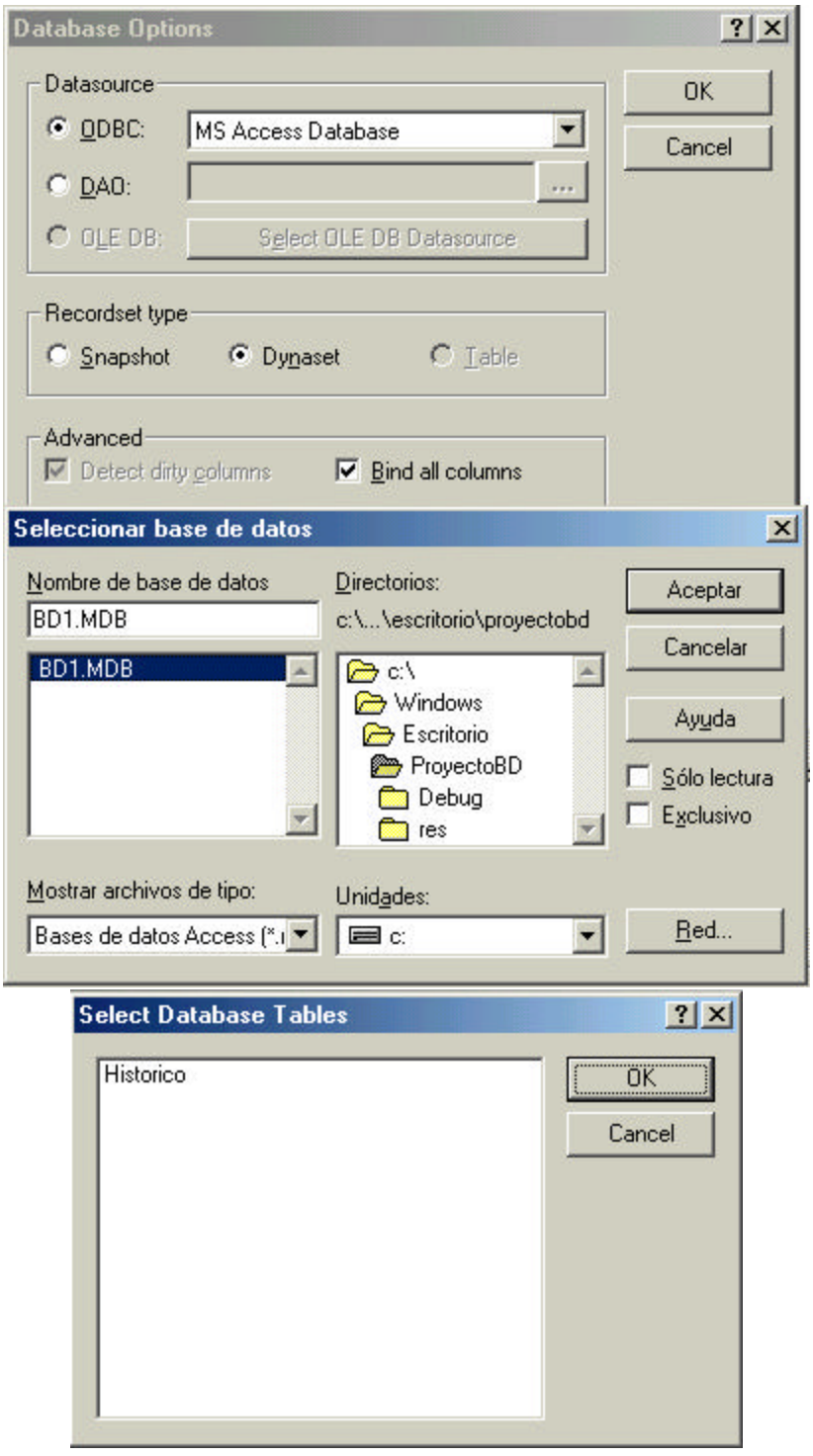
Para acceder a la base de datos local, previamente creada con Microsoft Access (*.mdb), utilizaremos la clase CRecordSet.

Para introducir la clase CRecordSet en una aplicación Visual C++ realizaremos los siguientes pasos.

Menu Insert -> NewClass->MFC Class-> CBaseDatosSet->*BaseClass* CRecordSet ->ODBC->MS Access Database->Dynaset->BD1.mdb->Historico

Nota: La base de datos debe estar copiada en la carpeta del proyecto.





-En la función *GetDefaultConnect()* de la clase *CBaseDatosSet* cambiaremos:

return _T("ODBC;DSN=MS Access Database");

por

return _T("ODBC;DRIVER={MICROSOFT ACCESS DRIVER (.mdb)};DSN='';DBQ=C:\\windows\\escritorio\\ProyectoBD\\bd1.mdb");*

Ahora se especifica el controlador ODBC que utilizaremos y donde esta la base de datos que utilizamos (bd1.mdb)

Crearemos unas variables que enlazaran con la base de datos

-Crear dos variables

CBaseDatosSet m_BaseDatosDlgSet;
CBaseDatosSet m_pSet;*

```
CBaseDatosSet m_BaseDatosDlgSet;
CBaseDatosSet* m_pSet;

CBaseDatosDlg(CWnd* pParent = NULL);   // standard constructor
```

Las variables creadas son una instancia de la clase CBaseDatosSet que es hija de la clase CRecordSet y por la tanto hereda todas las propiedades, métodos y eventos de dicha clase.

Principales miembros de la clase CRecorSet

Data Members

m_hstmt	Contains the ODBC statement handle for the recordset. Type HSTMT.
m_nFields	Contains the number of field data members in the recordset. Type UINT.
m_nParams	Contains the number of parameter data members in the recordset. Type UINT.
m_pDatabase	Contains a pointer to the CDatabase object through which the recordset is connected to a data source.
m_strFilter	Contains a CString that specifies a Structured Query Language (SQL) WHERE clause. Used as a filter to select only those records that meet certain criteria.
m_strSort	Contains a CString that specifies an SQL ORDER BY clause. Used to control how the records are sorted.

Construction

CRecordset	Constructs a CRecordset object. Your derived class must provide a constructor that calls this one.
Open	Opens the recordset by retrieving the table or performing the query that the recordset represents.
Close	Closes the recordset and the ODBC HSTMT associated with it.

Recordset Attributes

CanAppend	Returns nonzero if new records can be added to the recordset via the AddNew member function.
CanBookmark	Returns nonzero if the recordset supports bookmarks.
CanRestart	Returns nonzero if Requery can be called to run the recordset's query again.
CanScroll	Returns nonzero if you can scroll through the records.
CanTransact	Returns nonzero if the data source supports transactions.
CanUpdate	Returns nonzero if the recordset can be updated (you can add, update, or delete records).
GetODBCFieldCount	Returns the number of fields in the recordset.
GetRecordCount	Returns the number of records in the recordset.
GetStatus	Gets the status of the recordset: the index of the current record and whether a final count of the records has been obtained.
GetTableName	Gets the name of the table on which the recordset is based.
GetSQL	Gets the SQL string used to select records for the recordset.
IsOpen	Returns nonzero if Open has been called previously.
IsBOF	Returns nonzero if the recordset has been positioned before the first record. There is no current record.
IsEOF	Returns nonzero if the recordset has been positioned after the last record. There is no current record.
IsDeleted	Returns nonzero if the recordset is positioned on a deleted record.

Recordset Update Operations

AddNew	Prepares for adding a new record. Call Update to complete the addition.
CancelUpdate	Cancels any pending updates due to an AddNew or Edit operation.
Delete	Deletes the current record from the recordset. You must explicitly scroll to another record after the deletion.
Edit	Prepares for changes to the current record. Call Update to complete the edit.
Update	Completes an AddNew or Edit operation by saving the new or edited data on the data source.

Recordset Navigation Operations

GetBookmark	Assigns the bookmark value of a record to the parameter object.
Move	Positions the recordset to a specified number of records from the current record in either direction.
MoveFirst	Positions the current record on the first record in the recordset. Test for IsBOF first.
MoveLast	Positions the current record on the last record or on the last rowset. Test for IsEOF first.
MoveNext	Positions the current record on the next record or on the next rowset. Test for IsEOF first.
MovePrev	Positions the current record on the previous record or on the previous rowset. Test for IsBOF first.
SetAbsolute	Position Positions the recordset on the record corresponding to the specified record number.
SetBookmark	Positions the recordset on the record specified by the bookmark.

Other Recordset Operations

Cancel	Cancels an asynchronous operation or a process from a second thread.
FlushResultSet	Returns nonzero if there is another result set to be retrieved, when using a predefined query.
GetFieldValue	Returns the value of a field in a recordset.
GetODBCFieldInfo	Returns specific kinds of information about the fields in a recordset.
GetRowsetSize	Returns the number of records you wish to retrieve during a single fetch.
GetRowsFetched	Returns the actual number of rows retrieved during a fetch.
GetRowStatus	Returns the status of the row after a fetch.
IsFieldDirty	Returns nonzero if the specified field in the current record has been changed.
IsFieldNull	Returns nonzero if the specified field in the current record is Null (has no value).
IsFieldNullable	Returns nonzero if the specified field in the current record can be set to Null (having no value).
RefreshRowset	Refreshes the data and status of the specified row(s).
Requery	Runs the recordset's query again to refresh the selected records.
SetFieldDirty	Marks the specified field in the current record as changed.
SetFieldNull	Sets the value of the specified field in the current record to Null (having no value).
SetLockingMode	Sets the locking mode to "optimistic" locking (the default) or "pessimistic" locking. Determines how records are locked for updates.
SetParamNull	Sets the specified parameter to Null (having no value).
SetRowsetCursorPosition	Positions the cursor on the specified row within the rowset.

Recordset Overridables

Check	Called to examine the return code from an ODBC API function.
CheckRowsetError	Called to handle errors generated during record fetching.
DoBulkFieldExchange	Called to exchange bulk rows of data from the data source to the recordset. Implements bulk record field exchange (Bulk RFX).
DoFieldExchange	Called to exchange data (in both directions) between the field data members of the recordset and the corresponding record on the data source. Implements record field exchange (RFX).
GetDefaultConnect	Called to get the default connect string.
GetDefaultSQL	Called to get the default SQL string to execute.
OnSetOptions	Called to set options for the specified ODBC statement.
SetRowsetSize	Specifies the number of records you wish to retrieve during a fetch.

Para facilitar la programación hemos creado unas funciones propias, incluidas en el fichero *ABRdb.h* que añadimos a la clase CBaseDatosSet que facilitan la programación de la base de datos en cualquier aplicación.

Principales funciones creadas:

ABRAbrirBD();

Función que abre la base de datos para poder acceder a los datos.

ABRCerrarBD();

Función que abre la base de datos para poder acceder a los datos.

ABRUltimoRegistro();

Función que hace que el puntero de la base de datos (m_pSet) apunte al último registro de la base de datos.

ABRPrimerRegistro();

Función que hace que el puntero de la base de datos (m_pSet) apunte al primer registro de la base de datos.

ABRSiguienteRegistro();

Función que hace que el puntero de la base de datos (m_pSet) apunte al siguiente registro de la base de datos.

ABRAnteriorRegistro();

Función que hace que el puntero de la base de datos (m_pSet) apunte al anterior registro de la base de datos.

ABRInsertarRegistro(CString Nombre,CString Estado,CString Hora);

Función que inserta un registro al final de la base de datos.

ABRBorrarRegistro(CString Nombre,CString Estado,CString Hora);

Función que borra el registro que apunta el puntero de la base de datos (m_pSet).

ABRActualizarRegistro(CString Nombre,CString Estado,CString Hora);

Función que actualiza el registro que apunta el puntero de la base de datos (m_pSet).

ABRBuscarRegistro(CString Nombre,CString Estado,CString Hora);

Función que selecciona los registros con los campos introducidos, el puntero apunta a los registros seleccionados.

ABRBuscarTodosLosRegistros();

Función que selecciona todos los registros, el puntero apunta a todos los registros.

ABRContarRegistros();

Función que devuelve el numero total de registros de la base de datos.

Fundamentos del Puerto Paralelo

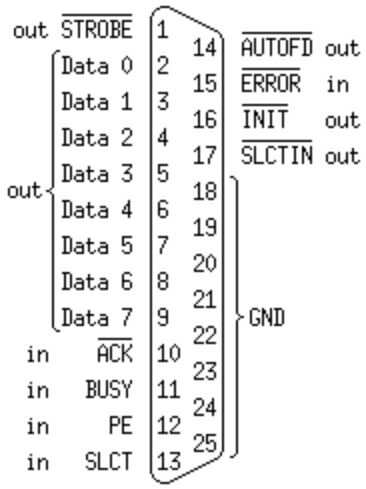
El puerto paralelo de una PC es ideal para ser usado como herramienta de control de motores, relés, LED's, etc. El mismo posee un bus de datos de 8 bits (Pin 2 a 9) y muchas señales de control, algunas de salida y otras de entrada que también pueden ser usadas fácilmente.

Las PC's generalmente poseen solo uno de estos puertos (LPT1) pero con muy poco dinero se le puede adicionar una tarjeta con un segundo puerto paralelo (LPT2).

En reglas generales la dirección hexadecimal del puerto LPT1 es igual a 0x378 (888 en decimal) y 0x278 (632 en decimal) para el LPT2. Esto se puede verificar fácilmente en el setup de la PC o bien en el cartel que generalmente la PC muestra en el momento del arranque. Puede darse el caso que el LPT1 asuma la dirección 0x3BC (956 en decimal) y el LPT2 0x378, en ese caso habrá que tratar de corregir el setup y/o los jumper de las tarjetas en caso que sea posible. De lo contrario se puede modificar el software para aceptar esas direcciones.

Breve descripción del puerto paralelo:

El puerto paralelo de un PC posee un conector de salida del tipo DB25 hembra cuyo diagrama y señales utilizadas podemos ver en la siguiente figura:



Si deseamos escribir un dato en el bus de salida de datos (pin 2 a 9) solo debemos escribir el byte correspondiente en la dirección hexadecimal 0X378 (888 en decimal) cuando trabajamos con el LPT1 y 0x278 (632 en decimal) cuando trabajamos con el LPT2. Los distintos pins (bits) de salida correspondientes al bus de datos no pueden ser escritos en forma independiente, por lo que siempre que se desee modificar uno se deberán escribir los ocho bits nuevamente.

Para leer el estado de los pins de entrada (10, 12, 13 y 15) se debe realizar una lectura a la dirección hexadecimal 0x379 (889 en decimal) si trabajamos con el LPT1 o bien leer la dirección 0x279 (633 en decimal) si trabajamos con el LPT2. La lectura será devuelta en un byte en donde el bit 6 corresponde al pin 10, el bit 5 corresponde al pin 12, el bit 4 corresponde al pin 13 y el bit 3 corresponde al pin 15.

En la siguiente tabla se puede ver lo antedicho en una forma más gráfica:

Escritura: Salida de Datos								
Escritura en dirección 0x378 (LPT1) o 0x278 (LPT2)								
DATO	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
DB25	Pin 9	Pin 8	Pin 7	Pin 6	Pin 5	Pin 4	Pin 3	Pin2

Lectura: Entrada de Datos								
Lectura en dirección 0x379 (LPT1) o 0x279 (LPT2)								
DATO	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
DB 25	No usar	Pin 10	Pin 12	Pin 13	Pin 15	No usar	No usar	No usar

Implementación de la clase CPuertoParalelo

Para utilizar el puerto paralelo con Visual C++ hemos implementado una clase llamada CPuertoParalelo que proporciona los métodos adecuados para controlar el puerto paralelo desde nuestra aplicación.

Esta clase esta formada por dos ficheros *PuertoParalelo.h* y *PuertoParalelo.cpp*

Los principales métodos que ofrece las clase son:

Leer()	Realiza la lectura de las entradas.
Escribir(int dato)	Realiza la escritura de las salidas.
Salida*n*(bool estado)	Realiza el cambio de estado de la salida *n*.
Entrada*n*()	Realiza la lectura de la entrada *n*.

Funcionamiento de la aplicación

La aplicación creada se encarga de controlar ocho salidas y cuatro entradas digitales del puerto paralelo, realizar una base de datos con la información de los eventos que se han producido y funcionar como una aplicación servidor para establecer una comunicación con otra aplicación igual instalada en un ordenador conectado en la misma red.

El programa esta formado por cuatro diálogos:

- El diálogo principal. Aplicación servidor. Que se encarga de controlar el puerto paralelo y de realizar las funciones de servidor.

- El diálogo de base de datos. Que gestiona la base de datos donde se registran todos los eventos.

- El diálogo Propiedades. Donde se configura la conexión y la Leyenda de las entradas y salidas.

- El diálogo cliente. Que realiza la función de cliente para conectarse con el diálogo servidor en la dirección IP y Puerto que le asignemos.

Con el programa incluimos una base de datos realizada con Microsoft Access para registrar los eventos que se producen. *(bd1.mdb)*

La base de datos dispone de una tabla llamada histórico formada por tres campos de texto:

- Nombre, donde se registra el nombre del evento que se ha producido.
- Estado, donde se registra el estado final del evento que se ha producido.
- Hora, donde se registra la hora en que se ha producido el evento.

La aplicación realizada dispone de las funciones servidor y cliente integrado todo en un mismo programa.

El programa servidor se ejecuta desde la computadora que se quiere controlar el puerto paralelo, pero también se instala en la computadora que se utiliza como cliente, aunque en este caso no se utilizan sus funciones de control del puerto y sólo se lanza el diálogo cliente para que la aplicación funcione como cliente del servidor remoto.

De esta manera una única aplicación se puede instalar en todos los ordenadores que se quiera y se pueden utilizar todos tanto como servidores como clientes.

Dialogo principal. Aplicación servidor

- Al arrancar la aplicación la primera función que se ejecuta es la OnInitDialog() de la clase CProyectoBDDlg

  Se establece un Timer que se encargara de ejecutar la función OnTimer() cada 500 ms que se encarga de comprobar si se ha producido algún cambio en las entradas digitales del puerto.

  Se abre la base de datos

  Se inicializan las salidas con los últimos estados registrados en la base de datos.

  Se actualiza el dialogo y las salidas del puerto.

  Se inserta un registro en la base de datos con la información de apertura de la aplicación.

  Se cierra la base de datos.

  Se captura del registro de Windows la información de la leyenda que aparece en el diálogo.

  Se pone el socket en modo de escuchar en la dirección de puerto que se indica en el diálogo.

Ahora la aplicación queda a la espera de que el usuario realice alguna acción sobre el dialogo del proyecto, sobre alguna opción del menú o se produzca algún cambio en las entradas del puerto, que se detecta automáticamente con la acción OnTimer()

- Si se actúa sobre algún CheckBox de las Salidas o alguna opción del menú de salidas.

  Se ejecutan las funciones OnCheckn() o OnSalidasSalidan() donde se inserta un registro en la base de datos con la información de la salida activada y se actualiza el puerto paralelo con la nueva información. Si el servidor estuviera conectado a algún cliente le enviaría el nuevo estado de todas las entradas y salidas.

- Si se actúa con la opción Base de Datos del menú Acción.

  Se ejecuta la función OnAccinBasededatos() donde se abre el diálogo de gestión de la Base de Datos de modo no modal (Se puede acceder a diálogo padre sin cerrar el propio diálogo)

- Si se actúa con la opción Propiedades del menú Acción

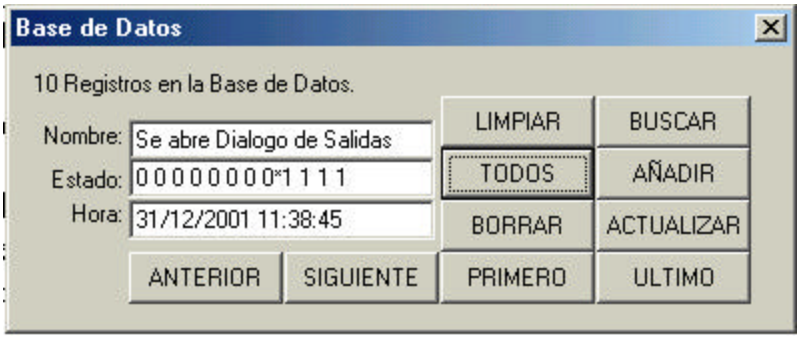
  Se ejecuta la función OnAccinPropiedades() donde se abre el diálogo de modificación de propiedades del proyecto de modo modal (No se puede acceder a diálogo padre sin cerrar el propio diálogo).En el caso de realizar alguna modificación, se actualizan los datos del diálogo principal

- Si se actúa con la opción Cliente del menú Acción

  Se ejecuta la función OnAccinCliente() donde se abre el diálogo cliente que funciona de modo modal, que funciona como cliente de cualquier otra aplicación servidor conectada a la red.

- Al cerrar la aplicación se ejecuta la función On Destroy() que registra el estado de las entradas y salidas y cierra la base de datos.

Diálogo Base de Datos



- Al arrancar el diálogo la primera función que se ejecuta es la OnInitDialog() de la clase CPropiedades1Dlg.

  Se abre la base de datos y se buscan todos los registros que contiene.

- Luego se ejecuta la función OnPaint() que escribe el último registro en los EditBox.

Ahora la aplicación queda a la espera de que el usuario realice alguna acción sobre el dialogo del Propiedades.

- Sin se actúa sobre algún Botón del diálogo ejecuta la función OnButtonX() y se realiza la función que indica.

- Si se modifica el contenido de algún EditBox se puede actualizar el registro correspondiente utilizando el botón que corresponde después de la modificación.

En cualquier momento se puede regresar al diálogo principal dejando es diálogo abierto o cerrarse como cualquier ventana de Windows

Diálogo Propiedades

- Al arrancar el diálogo la primera función que se ejecuta es la OnInitDialog() de la clase CPropiedades1Dlg

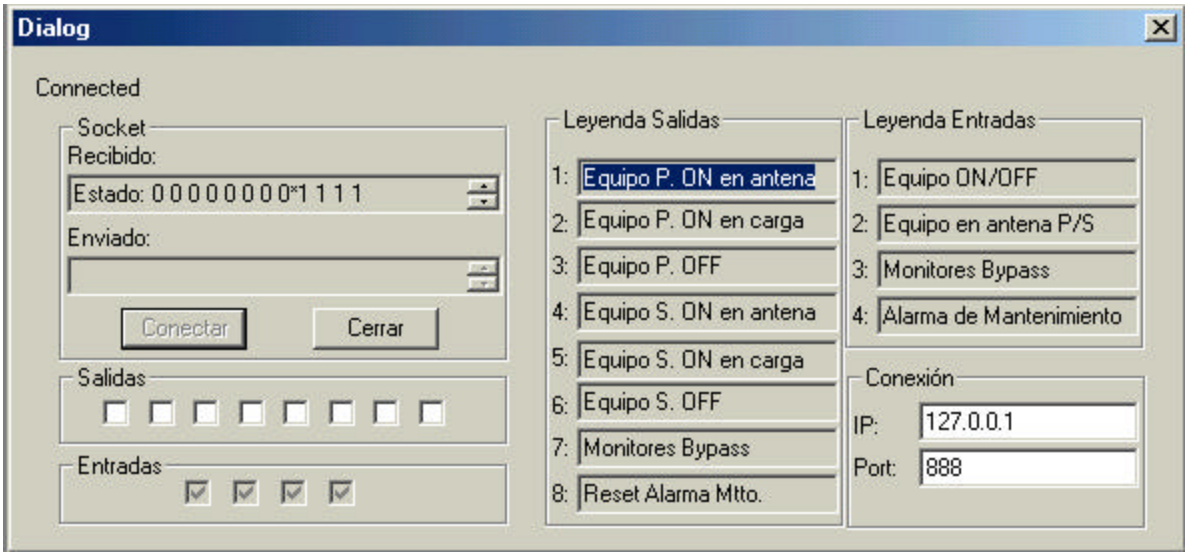
  Se captura del registro de Windows la información de la leyenda que aparece en el diálogo.

Ahora la aplicación queda a la espera de que el usuario realice alguna acción sobre el diálogo de Propiedades.

- Se puede modificar el contenido de los EditBox y al pulsar el Botón OK se ejecuta la función OnOk() que se encarga de escribir en el registro de Windows la información modificada y se regresa al diálogo principal.

- Si se cerrara el diálogo sin pulsar Ok no se registrarían las modificaciones realizadas en el registro de Windows.

Diálogo Cliente



- Al arrancar el diálogo la primera función que se ejecuta es la OnInitDialog() de la clase CClienteDlg.

  Se captura del registro de Windows la información de la conexión, IP, Port, que aparece en el diálogo.

Ahora la aplicación queda a la espera de que el usuario realice alguna acción sobre el diálogo Cliente.

- Se puede modificar el contenido de los EditBox de IP y Port para indicar en que dirección y en que puerto se debe realizar la conexión con el servidor. Al realizar cualquier cambio en los EditBox se ejecuta las funciones OnChangeEdit13() o OnChangeEdit14() que registran en el registro de Windows los nuevos datos actualizados.

- Al pulsar el Botón Conectar se ejecuta la función OnButtonConectar() que se encarga de realizar la conexión con el diálogo servidor.

- Al producirse la conexión la aplicación cliente empieza a recibir datos automáticamente por el Socket, se ejecuta la función OnDataArrivalWinsock1() y se recibe el estado de las salidas y entradas en el diálogo cliente y se inicializa la leyenda con la misma leyenda que tiene el diálogo servidor.

- Cada vez que se produce un cambio del estado de las entradas o salidas en el diálogo servidor y este está conectado a un diálogo cliente, se ejecuta la función DataArrivalWinsock1() y se recibe el nuevo estado actualizado.

- Si en el diálogo cliente se cambia el estado de algún CheckBox de las salidas se envía al servidor, a través del socket, una orden de cambio de la salida correspondiente en el servidor y queda registrada en la base de datos del servidor como si se hubiera ejecutado desde él mismo.

- Si durante la conexión se pulsa el Botón Cerrar se cierra la conexión del cliente con el servidor y se puede volver a conectar otra vez con el mismo servidor o con otro diferente en otra IP y puerto distinto.

- Si se cerrara el diálogo también se cierra la conexión y se regresa al diálogo principal.

Solución al problema de la IP dinámica

La dirección IP es el identificador de cada host dentro de su red. Cada host conectado a una red tiene una dirección IP asignada, la cual debe ser distinta a todas las demás direcciones que estén vigentes en ese momento en el conjunto de redes visibles por el host.

En el caso de Internet, no puede haber dos ordenadores con 2 direcciones IP (públicas) iguales. Pero sí podríamos tener dos ordenadores con la misma dirección IP siempre y cuando pertenezcan a redes independientes entre sí (sin ningún camino posible que las comunique).

Las direcciones IP se clasifican en:

-*Direcciones IP públicas*. Son visibles en todo Internet. Un ordenador con una IP pública es accesible (visible) desde cualquier otro ordenador conectado a Internet. Para conectarse a Internet es necesario tener una dirección IP pública.

-*Direcciones IP privadas (reservadas).* Son visibles únicamente por otros hosts de su propia red o de otras redes privadas interconectadas por routers. Se utilizan en las empresas para los puestos de trabajo. Los ordenadores con direcciones IP privadas pueden salir a Internet por medio de un router (o *proxy*) que tenga una IP pública. Sin embargo, desde Internet no se puede acceder a ordenadores con direcciones IP privadas.

A su vez, las direcciones IP pueden ser:

-*Direcciones IP estáticas (fijas).* Un host que se conecte a la red con dirección IP estática siempre lo hará con una misma IP. Las direcciones IP públicas estáticas son las que utilizan los servidores de Internet con objeto de que estén siempre localizables por los usuarios de Internet. Estas direcciones hay que contratarlas.

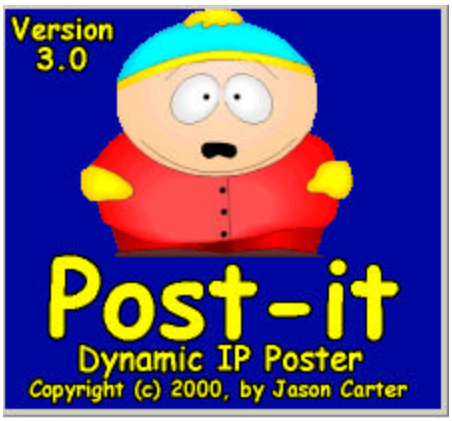
-*Direcciones IP dinámicas.* Un host que se conecte a la red mediante dirección IP dinámica, cada vez lo hará con una dirección IP distinta. Las direcciones IP públicas dinámicas son las que se utilizan en las conexiones a Internet mediante un módem. Los proveedores de Internet utilizan direcciones IP dinámicas debido a que tienen más clientes que direcciones IP (es muy improbable que todos se conecten a la vez).

Habitualmente, los ISP (*Internet Service Provider*) asignan en cada conexión una dirección IP dinámica. Esta dirección está activa para nosotros exclusivamente durante la sesión establecida con el ISP desde el momento de la conexión hasta la desconexión. Esta dirección IP nos permite, por supuesto, poder acceder a toda la red.

Si el PC donde ejecutamos nuestra aplicación servidor dispone de una dirección IP pública estática (Internet) o privada (intranet) no existe problema para realizar la conexión ya que sabemos en todo momento cual es la dirección IP del PC servidor.

El problema aparece si la dirección IP publica es dinámica, asignada por el proveedor de servicios de Internet, entonces el usuario del ordenador cliente puede no saber cual es la dirección IP del PC que actúa como servidor.

La solución al problema de la IP dinámica que se ha adoptado es la utilización de un software (Pos_it) que publica en Internet una pagina web donde puedes saber si el PC servidor esta conectado a la red y cual es su dirección IP pública.



Este software se debe arrancar y entonces se queda residente en el PC una vez ya existe conexión a Internet y ha sido asignada una IP pública dinámica por el proveedor. El programa se encarga de publicar cada cierto tiempo una página web con la información requerida actualizada.

The following is my current IP: 194.224.211.92

Post-it started on 25/12/2001 at 10:42:25
This page is being updated every 1 minutes
Last upload was on 25/12/2001 at 10:57:27
The number of uploads for this session is 16
Post-it: Dynamic IP Poster
Copyright © 1999, Jason Carter

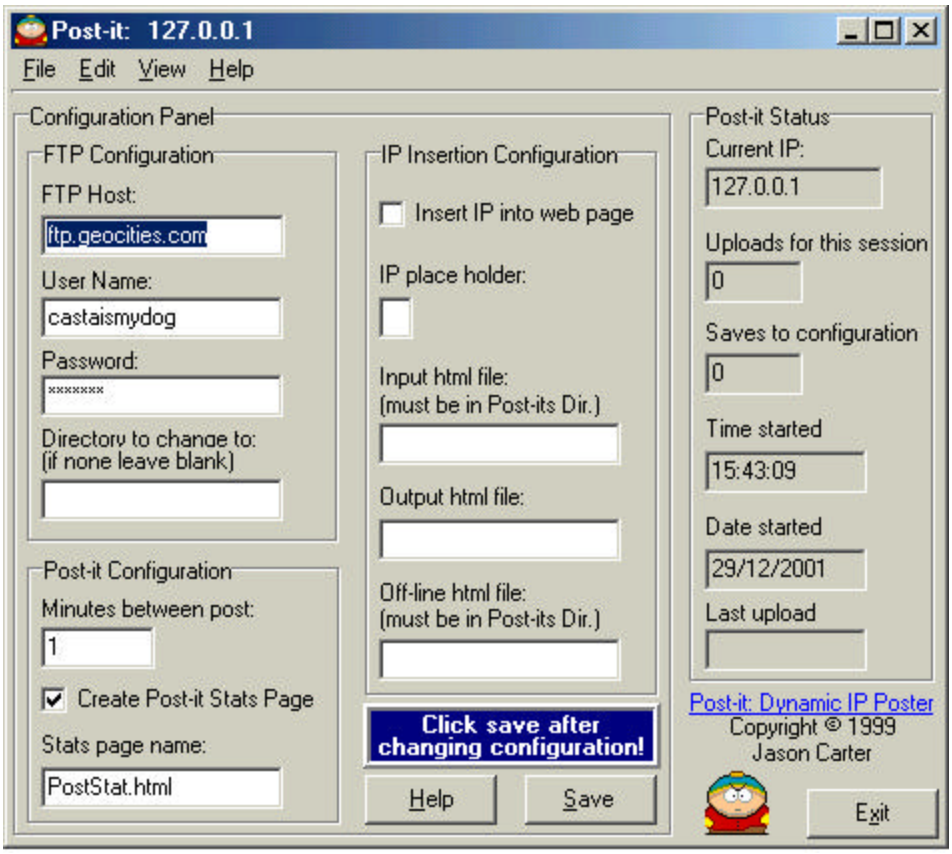
En el momento que desaparece la conexión a Internet la página web avisa que el PC no esta conectado

I am currently offline!!!

Post-it: Dynamic IP Poster
Copyright © 1999, Jason Carter

La configuración de este software es muy simple y admite la opción de introducir la dirección IP pública en una página web ya creada.

La pantalla de configuración es la siguiente



FTP Host indica el servidor FTP donde va publicarse la página web con la información requerida

User Name indica el nombre de usuario de la sesión FTP

Password indica la clave de acceso al servidor FTP

Minutes between post indica los minutos entre publicación de páginas web actualizadas

Create Post-it Stats Page indica si se quiere crear una página web estándar con la información requerida

Stats page name indica el nombre de la página web que se publicará

Conclusiones

Con el presente proyecto hemos realizado una aplicación informática para ejecutar el control y monitorización remota de unos equipos electrónicos aeronáuticos desde un PC conectado en una red Ethernet, capaz de controlar el puerto paralelo del PC, que se encuentra en la instalación de la radioayuda y de comunicarse con otra aplicación situada en el lugar donde se centralizan la monitorización de todas las instalaciones.

Mediante el puerto paralelo del PC de la instalación disponemos de ocho salidas y cuatro entradas digitales con las cuales podremos generar las ordenes de control pertinentes y monitorizar el estado de la instalación.

También realizamos un registro sobre una base de datos de todos los eventos que producen las entradas y salidas, necesario para la buena ejecución de las tareas de mantenimiento de las instalaciones a controlar.

Se ofrece tres tipos de configuraciones hardware para realizar la conexión, aunque no hay que olvidar que existe la posibilidad de adapatarse a cualquier red basada en TCP/IP.

Se realiza una breve explicación de fundamentos sobre TCP/IP, el control winsock, bases de datos y el puerto paralelo del PC y la necesidad de su utilización en este proyecto.

Añadimos un apartado donde se explica como funciona la aplicación desde el punto de vista de programación, que ayudará a entender el código de la aplicación.

Se plantea una posible solución al problema de la IP dinámica sin olvidar el gran número de soluciones que se podrían plantear.

En la memoria de diseño se pretende dar una guía de cómo realizar una base de datos que se adapte a nuestra aplicación y otra guía de cómo realizar la aplicación paso a paso. También se incluye el código de la funciones en C++ realizadas y una guía de instalación del software.

Por último se ha realizado un breve manual de usuario para facilitar su utilización a un usuario sin conocimientos de programación.

En el anexo añadimos el código de todos los archivos que se incluyen en el proyecto de Microsoft Visual C++

Posibles líneas de continuación del Proyecto

Este proyecto realiza una introducción a la comunicación entre aplicaciones con socket's. A partir de aquí se abre un amplio campo de posibilidades para utilizar esta funcionalidad en múltiples aplicaciones.

A nivel de comunicación mediante socket's en nuestra aplicación hemos utilizado solo una conexión punto a punto entre aplicaciones pero existen otras posibilidades que podrían ser de utilidad para mejorar la comunicación.

- Un socket para comandos y otro socket para datos.

- Varios sockets para conexiones entrantes de varias aplicaciones cliente al mismo servidor y otro para los datos.

- En visual C++ existe la posibilidad de crear controles en tiempo de ejecución con lo que podríamos crear un socket nuevo con cada conexión entrante.

- Otra posibilidad muy interesante sería crear una aplicación servidor con la única funcionalidad de gestionar los mensajes provenientes de las diferentes aplicaciones cliente que se le conectan. Esta posibilidad presenta la ventaja de que se puede instalar la aplicación servidor en una computadora que su dirección IP sea fija y entonces todas las aplicaciones cliente saben donde dirigirse para establecer la comunicación y es la aplicación servidor quien se encargaría de reenviar la información a la aplicación cliente de destino en el caso de estar conectada.

- Utilizando socket's se pueden crear aplicaciones cliente que en lacen con servidores de POP3 y SMTP (email), http (páginas web), FTP (Transferencia de ficheros),etc. con lo que se puede añadir funciones como alertas por correo electrónico a tu buzón de correo, móvil, etc. Presentación de información de la aplicación en una página web, transferencia de las bases de datos en otro ordenador, etc.

- Los socket's pueden ser utilizados con otros lenguajes de programación como ASP que es un lenguaje de programación para realizar páginas web que se ejecutan sobre la maquina servidor. Con esto se puede conseguir que todo el potencial que presentan los socket's se pueda utilizar desde cualquier máquina con cualquier sistema operativo con posibilidad de navegar por Internet o intranet, indiferentemente del navegador que utilice.

- Posibilidad de utilización de password's de acceso a la aplicación para que solo usuarios registrados puedan conectarse y tener acceso a la aplicación.

- Posibilidad de encriptar la información que enviamos a la red para evitar posibles usos de información capturada desde la red.

A nivel de funcionalidad de la aplicación servidor en nuestro proyecto se ha optado por el control del puerto paralelo con lo que ya se obtienen una amplia gama de funciones en muchos campos.

- Se podría añadir el control de alguna cámara de video que publique las imágenes en Internet. A través de nuestra aplicación podríamos controlar la dirección, selección, zoom, etc.

- Otra posibilidad con lo que se aumentaría mucho la capacidad de nuestras aplicaciones sería la comunicación mediante RS232 de nuestra aplicación con un microcontrolador externo o un PLC. Con esto podríamos aprovechar todas las posibilidades del microcontrolador elegido y controlarlas desde una aplicación en un PC remoto. A partir de aquí se puede realizar una red IIC o inalámbrica de microcontroladores esclavos con un microcontrolador master conectado al PC.

- Una opción muy interesante sería la de poder controlar el ordenador servidor telefónicamente para poder arrancarlo, conectarse a la red o arrancar la aplicación cliente-servidor que se desee.

- Por último, ya que la lista de mejoras podría continuar indefinidamente, añadiría la posibilidad de crear aplicaciones cliente-servidor mediante microcontroladores y así no utilizar un PC que es mucho más caro. Las aplicaciones servidor se implementan en un chip (micro servidor) que dispone de una pila TCP/IP y de un modem o una adaptador de red para conectarse a la red. El circuito integrado estará conectado a la red y podrá utilizar socket's, emails, páginas web, etc y a partir de aquí conectarse con PC o con otros microservidores que realizan las funciones necesarias en cada caso. Con esto podríamos añadir a cualquier electrodoméstico, a cualquier electroválvula, a cualquier sensor funcionalidad cliente-servidor y ser controlado desde cualquier punto de la red en los que se conecten.

UNIVERSITAT ROVIRA I VIRGILI

Proyecto Final de Carrera:

CONTROL REMOTO MEDIANTE COMUNICACIÓN TCP/IP DE RADIOAYUDAS AERONAUTICAS

2.MEMORIA DISEÑO

- *Ingeniero Superior en Automática y Electrónica Industrial* — Agustín Baubí Rovira
- *Profesor Ponente* — Ernest Gil Dolcet

Febrero 2002

INDICE

Realizacion de la base de datos del proyecto.....1
Ejemplo de Base de datos 2
Realización del programa de base de datos en Visual C++ 3
Crear y preparar la aplicación principal del proyecto (ProyectoBD) 3
Preparar el proyecto para utilizar funciones de Visual C++ sobre bases de datos 4
Crear diálogo para ver la Base de Datos.....5
Crear Menu para el diálogo principal.....6
En la clase CProyectoBDDlg crear función OnAccinBasededatos() 7
CASO NO MODAL 7
CASO MODAL 9
Realizar CTRL+F5 para compilar hasta aquí el proyecto 9
Crear la clase RecorSet para acceder a la Base de Datos 10
Preparar la clase CBaseDatosSet para añadir funciones propias 12
Preparar CBaseDatosDlg para utilizar la clase RecordSet 13
Completar diálogo de la base de datos 14
Preparar la clase CBaseDatosDlg para utilizar funciones propias.....16
Preparar CProyectoBDDlg para utilizar el RecordSet.....22
Realizar CTRL+F5 para compilar hasta aquí el proyecto 22
Completar diálogo principal del Proyecto 23
Añadir a CProyectoBDDlg la clase para controlar el puerto paralelo 34
Añadir a CProyectoBDDlg funciones de servidor.....35
Preparar la clase CMSWinsockControl para utilizar funciones propias 37
Realizar CTRL+F5 para compilar hasta aquí el proyecto 39
Añadir a CProyectoBDDlg un dialogo de propiedades 40
Realizar CTRL+F5 para compilar hasta aquí el proyecto 44
Añadir a CProyectoBDDlg un diálogo cliente 45
Preparar la clase CClienteDlg para utilizar funciones propias 47
Realizar CTRL+F5 para compilar hasta aquí el proyecto 54
Ficheros que incluimos en el proyecto 55
PuertoParalelo.cpp y PuertoParalelo.h, clase que controla el puerto paralelo 56
ABRdb.h, funciones para gestionar la base de datos.....58
ABRCBaseDatosDlgUtils.h, funciones útiles para el diálogo base de datos 61
ABRCProyectoBDDlgUtils.h, funciones útiles para el diálogo principal.....62
ABRWinsock.h, funciones útiles gestionar el control Winsock.....69
ABRCClienteDlgUtils.h, funciones útiles para el diálogo cliente 72
Instalación de la aplicación ProyectoBD.exe 78

Realizacion de la base de datos del proyecto

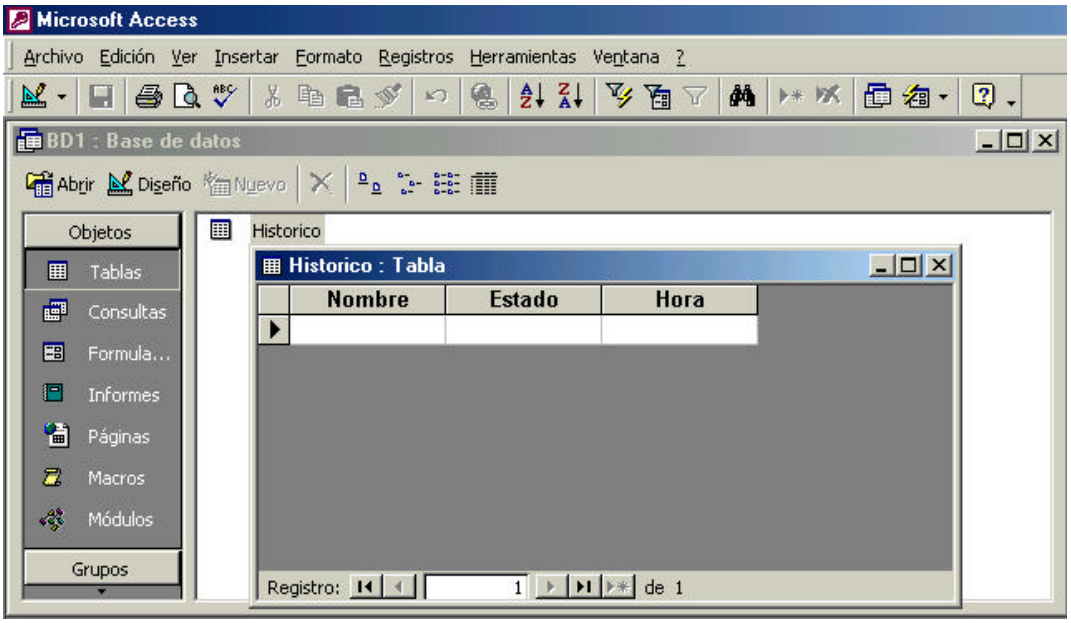
Utilizaremos una base de datos creada en Microsoft Access para realizar un histórico de los diferentes sucesos que se producen durante la ejecución de la aplicación.

La base de datos esta formado por una tabla llamada Historico con tres campos.

Nombre, campo donde se registran el tipo de suceso que se ha producido.

Estado, campo donde se registra el estado final de la o las entradas y salidas al producirse el suceso.

Hora, campo donde se registra la hora en la cual se registra el suceso.



Los diferentes sucesos que pueden aparecer son:

Inicio de la Base de datos
Cuando se realiza el primer registro de la base de datos.

Se abre diálogo de Salidas

Cuando se ejecuta la aplicación del proyecto.
En el campo estado se registra el estado en el que se activan todas las salidas y entradas.

Se cierra diálogo de Salidas

Cuando se cierra la aplicación del proyecto.
En el campo estado se registra el estado en el que se quedan todas las salidas y entradas.

Entradan

Cuando cambia el estado actual de una entrada del puerto paralelo.
Siendo n el número de la entrada que ha cambiado.
En el campo estado se registra el estado 1 o 0 que se encuentra esta entrada.

Salidan

Cuando cambia el estado actual de una salida del puerto paralelo.
Siendo n el número de la salida que ha cambiado.
En el campo estado se registra el estado 1 o 0 que se encuentra esta salida.

En todos los suceso se registra en el campo hora la hora y el dia en que se ha producido el nuevo estado

Ejemplo de Base de dat os

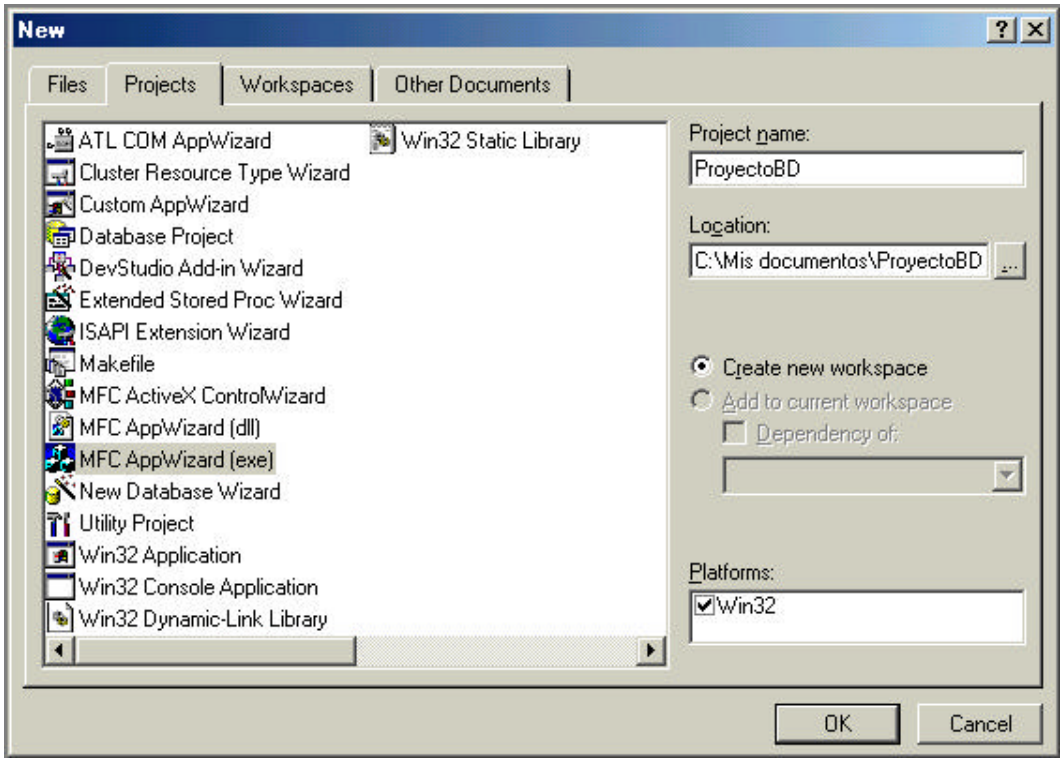
Nombre	Estado	Hora
Inicio de la Base de Datos	-	24/09/2001 18:47:53
Se abre Dialogo de Salidas	0 0 0 0 0 0 0 0*0 0 0 0	24/09/2001 18:47:53
Entrada1	1	24/09/2001 18:47:57
Entrada2	1	24/09/2001 18:47:57
Entrada3	1	24/09/2001 18:47:57
Entrada4	1	24/09/2001 18:47:57
Se Cierra Dialogo de Salidas	0 0 0 0 0 0 0 0*1 1 1 1	24/09/2001 18:48:11
Se abre Dialogo de Salidas	0 0 0 0 0 0 0 0*1 1 1 1	15/12/2001 16:00:52
Salida1	1	15/12/2001 16:01:04
Salida2	1	15/12/2001 16:01:06
Salida3	1	15/12/2001 16:01:07
Salida4	1	15/12/2001 16:01:08
Salida5	1	15/12/2001 16:01:09
Se Cierra Dialogo de Salidas	1 1 1 1 1 0 0 0*1 1 1 1	15/12/2001 16:01:15
Se abre Dialogo de Salidas	1 1 1 1 1 0 0 0*1 1 1 1	15/12/2001 16:16:46
Entrada3	0	15/12/2001 16:16:46
Entrada4	0	15/12/2001 16:16:46
Salida6	1	15/12/2001 16:16:58
Salida7	1	15/12/2001 16:17:00
Salida8	1	15/12/2001 16:17:02
Entrada1	0	15/12/2001 16:17:13
Entrada2	0	15/12/2001 16:17:18
Entrada4	1	15/12/2001 16:17:23
Entrada3	1	15/12/2001 16:17:25
Salida8	0	15/12/2001 16:17:44
Salida7	0	15/12/2001 16:17:45
Salida6	0	15/12/2001 16:17:46
Salida5	0	15/12/2001 16:17:47
Salida4	0	15/12/2001 16:17:49
Salida3	0	15/12/2001 16:17:52
Salida2	0	15/12/2001 16:17:54
Salida8	1	15/12/2001 16:17:56
Se Cierra Dialogo de Salidas	1 0 0 0 0 0 0 1*0 0 1 1	15/12/2001 16:18:10
Se abre Dialogo de Salidas	1 0 0 0 0 0 0 1*0 0 1 1	15/12/2001 16:27:45
Se Cierra Dialogo de Salidas	1 0 0 0 0 0 0 1*0 0 1 1	15/12/2001 16:28:08

Realización del programa de base de datos en Visual C++

Crear y preparar la aplicación principal del proyecto (ProyectoBD)

-En el asistente de creación de una aplicación MFC aplicaremos las siguientes opciones:

-Aplicación badasa en dialogo.
-Librerías estaticas.



-Copiar en la carpeta del proyecto los ficheros que deberemos incluir en el proyecto para utilizar funciones propias útiles para las diferentes partes del proyecto.

PuertoParalelo.cpp	
PuertoParalelo.h	Clase que controla el puerto paralelo.
ABRdb.h	Funciones para gestionar la base de datos.
ABRCBaseDatosDlgUtils.h	Funciones útiles para el diálogo base de datos.
ABRCProyectoBDDlgUtils.h	Funciones útiles para el diálogo principal.
ABRWinsock.h	Funciones útiles gestionar el control Winsock.
ABRCClienteDlgUtils.h	Funciones útiles para el diálogo cliente.

-Preparar y copiar en la carpeta del proyecto el fichero de la Base de datos que utilizaremos como historico en la aplicación (*bd1.mdb*).

Tabla:	Historico	
Campos:	Nombre	Se registra que salida o entrada a cambiado.
	Estado	Se registra el estado en que se encuentra (On / Off).
	Hora	Se registra a que hora se ha producido el cambio de estado

Preparar el proyecto para utilizar funciones de Visual C++ sobre bases de datos

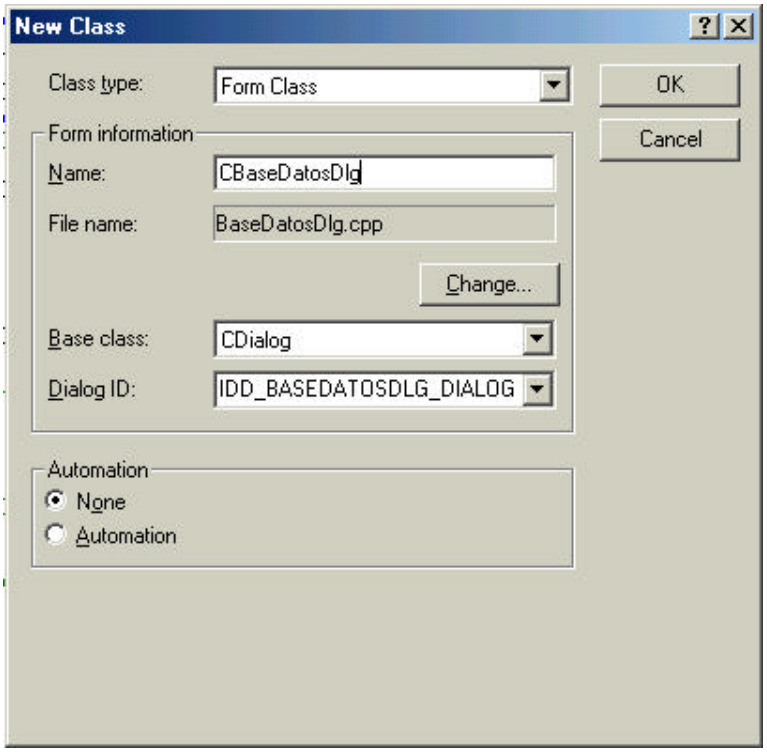
-Añadir funcionalidad ODBC.

Añadir a *StdAfx.h*:
#include <afxdb.h>
#include <afxdao.h>

```
// stdafx.h : include file for standard system include files,
//  or project specific include files that are used frequently, but
//      are changed infrequently
//

#if !defined(AFX_STDAFX_H__9210F9F8_A9F7_11D5_97E1_FCCD2DB3034F__INCLUDED_)
#define AFX_STDAFX_H__9210F9F8_A9F7_11D5_97E1_FCCD2DB3034F__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

#define VC_EXTRALEAN          // Exclude rarely-used stuff from Windows headers

#include <afxwin.h>         // MFC core and standard components
#include <afxext.h>         // MFC extensions
#include <afxdisp.h>        // MFC Automation classes
#include <afxdtctl.h>         // MFC support for Internet Explorer 4 Common Controls
#ifndef _AFX_NO_AFXCMN_SUPPORT
#include <afxcmn.h>                  // MFC support for Windows Common Controls
#endif // _AFX_NO_AFXCMN_SUPPORT

#include <afxdb.h>
#include <afxdao.h>

//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately before the
previous line.

#endif // !defined(AFX_STDAFX_H__9210F9F8_A9F7_11D5_97E1_FCCD2DB3034F__INCLUDED_)
```

Crear diálogo para ver la Base de Datos
(IDD_BASEDATOSDLG_DIALOG)

Insert->New Class->Form Class->CBaseDatosDlg

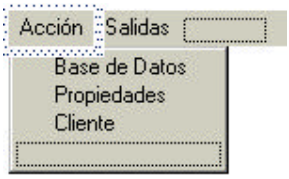


Crear Menu para el diálogo principal
(*IDR_MENU1*)

Insert->Resource->Menu

Acción->Base de Datos
Propiedades
Cliente

Salidas->Salida1
Salida2
...
Salida8



El Menu Base de Datos visualizará el diálogo que gestiona la base de datos.

El Menu Propiedades visualizará el diálogo que gestiona las propiedades del proyecto.

El Menu Cliente visualizará el diálogo que realiza la función de cliente para conectarse con el diálogo principal de la misma aplicación, servidor, o de otra aplicación igual que se ejecuta en otro ordenador conectado a la misma red.

-Asignar Menu al diálogo principal
(*IDD_PROYECTOBD_DIALOG*)

En el dialogo principal-> Propiedades -> Menu -> *IDR_MENU1*

En la clase CProyectoBDDlg crear función OnAccinBasede datos()

Esta función se encargará de visualizar el diálogo hijo *IDD_BASEDATOSDLG_DIALOG* que gestiona la base de datos.

Existen dos formas de visualizar el dialogo hijo:

Caso Modal: El diálogo se visualiza y no se puede regresar al otro diálogo hasta que cerramos el actual.
Caso No Modal: El diálogo se visualiza y si se puede regresar al otro diálogo, el diálogo hijo permanece en segundon plano.

En este caso el diálogo de gestion de la base de datos lo visualizaremos de forma no modal.

CASO NO MODAL

-En el fichero *BaseDatosDlg.cpp* (en la clase *CBaseDatosDlg*)

-Modificar la función *CBaseDatosDlg::CBaseDatosDlg((CWnd* pParent /*=NULL*/)* y incluir *Create(IDD);*

```
CBaseDatosDlg::CBaseDatosDlg(CWnd* pParent /*=NULL*/)            :
CDialog(CBaseDatosDlg::IDD, pParent)
{
        //{{AFX_DATA_INIT(CBaseDatosDlg)
        //}}AFX_DATA_INIT
        Create(IDD);
}
```

-En el fichero *ProyectoBDDlg.h*

-Incluir *class CBaseDatosDlg;* antes de la definición de la clase (*class CProyectoBDDlg : public CDialog)*

```
/////////////////////////////////////////////////////////////////////////////
// CProyectoBDDlg dialog
class CBaseDatosDlg;
class CProyectoBDDlg : public CDialog
{
```

-Crear Variable -> *CBaseDatosDlg* m_pBaseDatosDlg;*

```
public:
        CProyectoBDDlg(CWnd* pParent = NULL);       // standard constructor
        CBaseDatosDlg* m_pBaseDatosDlg;
        // Dialog Data
```

-En el fichero *ProyectoBDDlg.cpp*

-En el constructor *CProyectoBDDlg::CProyectoBDDlg(CWnd* pParent /*=NULL*/)*
Incluir *m_pBaseDatosDlg = new CBaseDatosDlg();*

```
CProyectoBDDlg::CProyectoBDDlg(CWnd* pParent /*=NULL*/)          :

CDialog(CProyectoBDDlg::IDD, pParent)
{
```

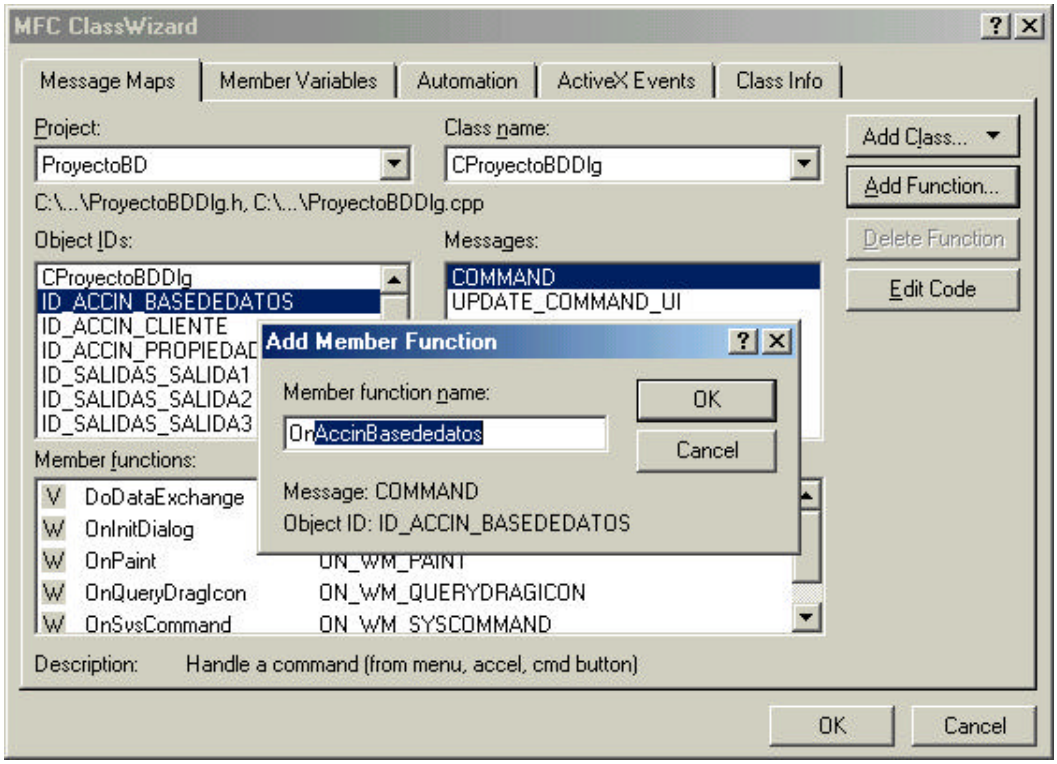
```
        //{{AFX_DATA_INIT(CProyectoBDDlg)
        //}}AFX_DATA_INIT
        // Note that LoadIcon does not require a subsequent DestroyIcon in Win32
        m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
        m_pBaseDatosDlg = new CBaseDatosDlg();
}
```

-Añadir *#include "BaseDatosDlg.h"*

```
// ProyectoBDDlg.cpp : implementation file
//

#include "stdafx.h"
#include "ProyectoBD.h"
#include "ProyectoBDDlg.h"

#include "BaseDatosDlg.h"
```

-Insertar función *OnAccinBasedeDatos()*

(Classwizard (Ctrl+W)-> MessageMap)
(ID_ACCIN_BASEDEDATOS -> COMMAND-> Add Function -> Ok ->EditCode)




```
void CProyectoBDDlg::OnAccinBasededatos()
{
// TODO: Add your command handler code here
m_pBaseDatosDlg->ShowWindow(SW_SHOW);
m_pBaseDatosDlg->SetFocus ();
}
```

Nota: Si se quisiera utilizar el caso modal se realizaría de la siguiente manera.

CASO MODAL

-En el fichero *ProyectoBDDlg.h*

-Crear Variable (Botón derecho sobre CProyectoBDDlg -> Add Member variable)

```
public:
	CProyectoBDDlg(CWnd* pParent = NULL);	// standard constructor

	CBaseDatosDlg BaseDatosDlg;
// Dialog Data
```

-Insertar función *OnAccinBasedeDatos()*

(Classwizard -> MessageMap)
(ID_ACCIN_BASEDEDATOS -> COMMAND-> Add Function ->
Ok ->EditCode)

```
void CProyectoBDDlg::OnAccinBasededatos()
{
// TODO: Add your command handler code here
BaseDatosDlg.DoModal ();
}
```

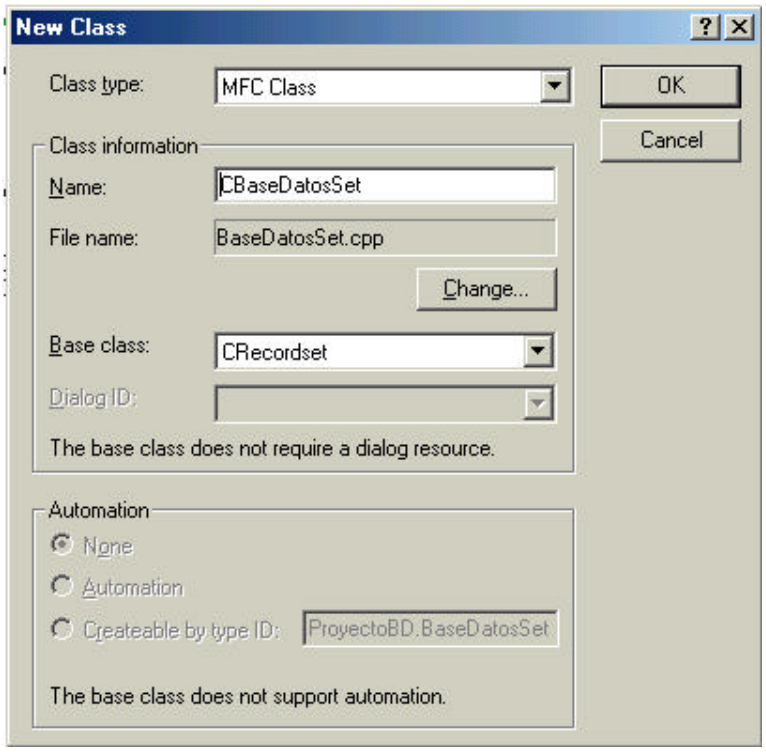
Realizar CTRL+F5 para compilar hasta aquí el proyecto

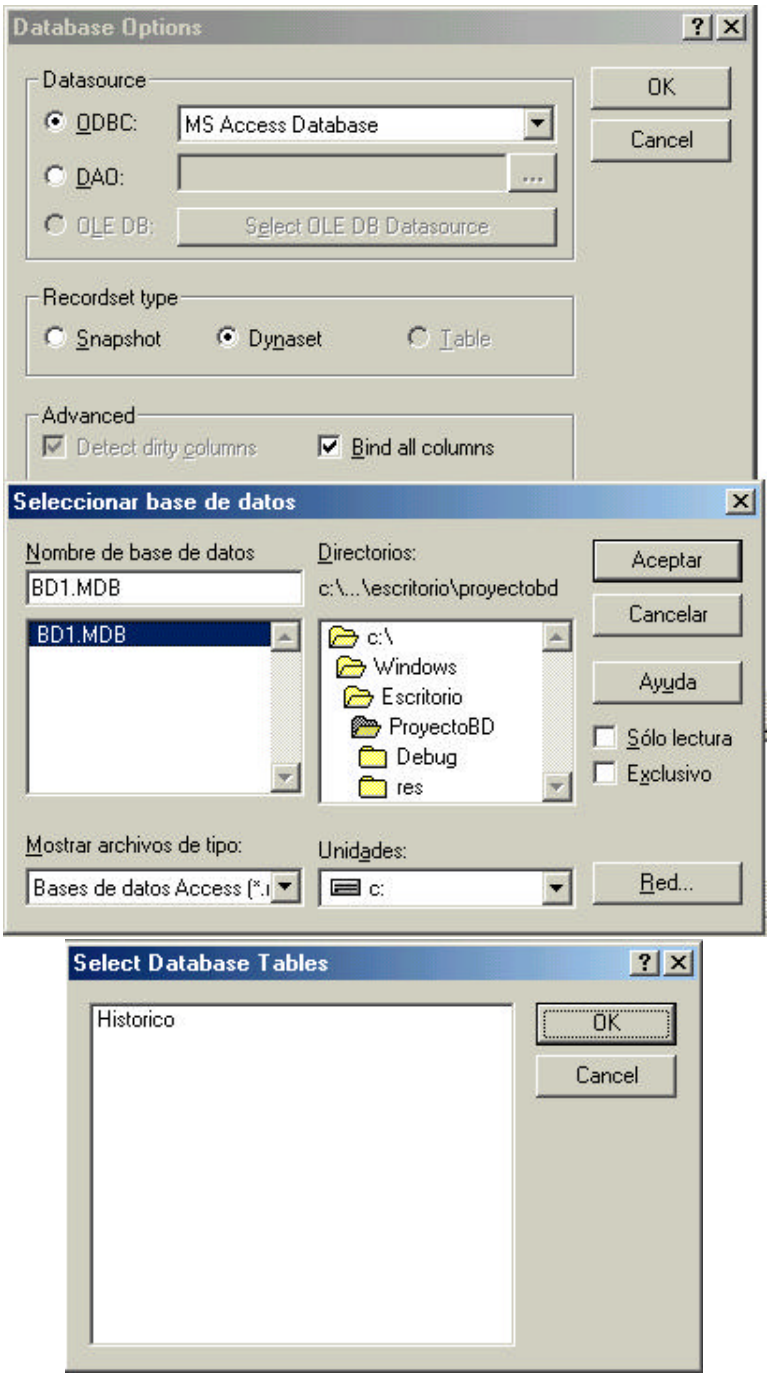
Crear la clase RecorSet para acceder a la Base de Datos

Para crear la clase realizar:

Insert -> NewClass->MFC Class-> CBaseDatosSet->*BaseClass* CRecordSet ->ODBC->MS Access Database->Dynaset->BD1.mdb->Historico

Nota: La base de datos debe estar copiada en la carpeta del proyecto.





-En la función *GetDefaultConnect()* de la clase *CBaseDatosSet* cambiar:

return _T("ODBC;DSN=MS Access Database");

por

return _T("ODBC;DRIVER={MICROSOFT ACCESS DRIVER (.mdb)};DSN='';DBQ=C:\\windows\\escritorio\\ProyectoBD\\bd1.mdb");*

```
CString CBaseDatosSet::GetDefaultConnect()
{
	return _T("ODBC;DRIVER={MICROSOFT ACCESS DRIVER
(*.mdb)};DSN='';DBQ=C:\\windows\\escritorio\\ProyectoBD\\bd1.mdb");
}
```

Preparar la clase CBaseDatosSet para añadir funciones propias

-Añadir a *BaseDatosSet.h* las declaraciones de las funciones del fichero *ABRdb.h* para gestionar la base de datos.

```
// Overrides
        // ClassWizard generated virtual function overrides
        //{{AFX_VIRTUAL(CBaseDatosSet)
        public:
        virtual CString GetDefaultConnect();    // Default connection string
        virtual CString GetDefaultSQL();    // Default SQL for Recordset
        virtual void DoFieldExchange(CFieldExchange* pFX);  // RFX support

//funciones Creadas por ABR
void CBaseDatosSet::ABRAbrirBD();
void CBaseDatosSet::ABRCerrarBD();
void CBaseDatosSet::ABRUltimoRegistro();
void CBaseDatosSet::ABRPrimerRegistro();
void CBaseDatosSet::ABRSiguienteRegistro();
void CBaseDatosSet::ABRAnteriorRegistro();
void CBaseDatosSet::ABRInsertarRegistro(CString Nombre,CString Estado,CString Hora);
void CBaseDatosSet::ABRBorrarRegistro(CString Nombre,CString Estado,CString Hora);
void CBaseDatosSet::ABRActualizarRegistro(CString Nombre,CString Estado,CString Hora);
void CBaseDatosSet::ABRBuscarRegistro(CString Nombre,CString Estado,CString Hora);
void CBaseDatosSet::ABRBuscarTodosLosRegistros();
CString CBaseDatosSet::ABRContarRegistros();
// fin de las declaraciones
        //}}AFX_VIRTUAL
```

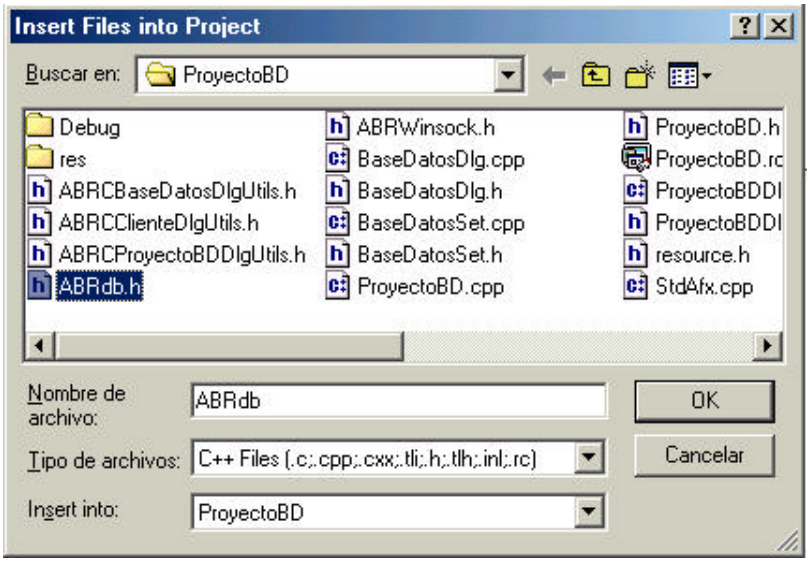
-Añadir a *BaseDatosSet.cpp*

#include "ABRdb.h"

```
// BaseDatosSet.cpp : implementation file
//

#include "stdafx.h"
#include "ProyectoBD.h"
#include "BaseDatosSet.h"
#include "ABRdb.h"
#ifdef _DEBUG
```

Debe estar copiado el fichero *ABRdb.h* en la carpeta del proyecto, fichero que incluye las funciones de gestión de la base de datos.

Project -> Add to Project -> Files -> *ABRdb.h*



Preparar CBaseDatosDlg para utilizar la clase RecordSet

-Añadir en *BaseDatosDlg.h*

#include "BaseDatosSet.h"
class CBaseDatosSet;

```
#if !defined(AFX_BASEDATOSDLG_H__9210F9FE_A9F7_11D5_97E1_FCCD2DB3034F__INCLUDED_)
#define AFX_BASEDATOSDLG_H__9210F9FE_A9F7_11D5_97E1_FCCD2DB3034F__INCLUDED_

#include "BaseDatosSet.h"          // Added by ClassView
#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000
// BaseDatosDlg.h : header file
//

/////////////////////////////////////////////////////////////////////////////
// CBaseDatosDlg dialog

class CBaseDatosSet;
class CBaseDatosDlg : public CDialog
```

-Crear dos variables

CBaseDatosSet m_BaseDatosDlgSet;
CBaseDatosSet m_pSet;*

```
CBaseDatosSet m_BaseDatosDlgSet;
CBaseDatosSet* m_pSet;

CBaseDatosDlg(CWnd* pParent = NULL);   // standard constructor
```

Completar diálogo de la base de datos

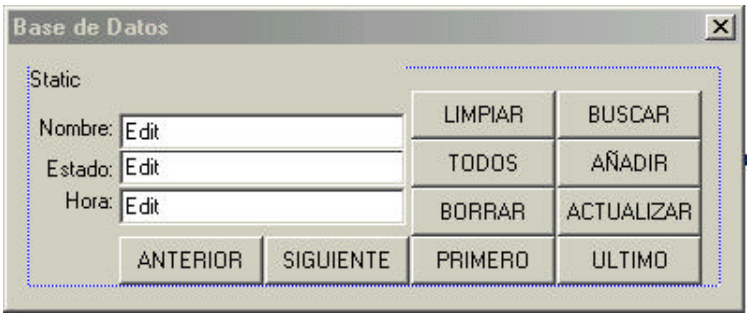
(IDD_BASEDATOSDLG_DIALOG)

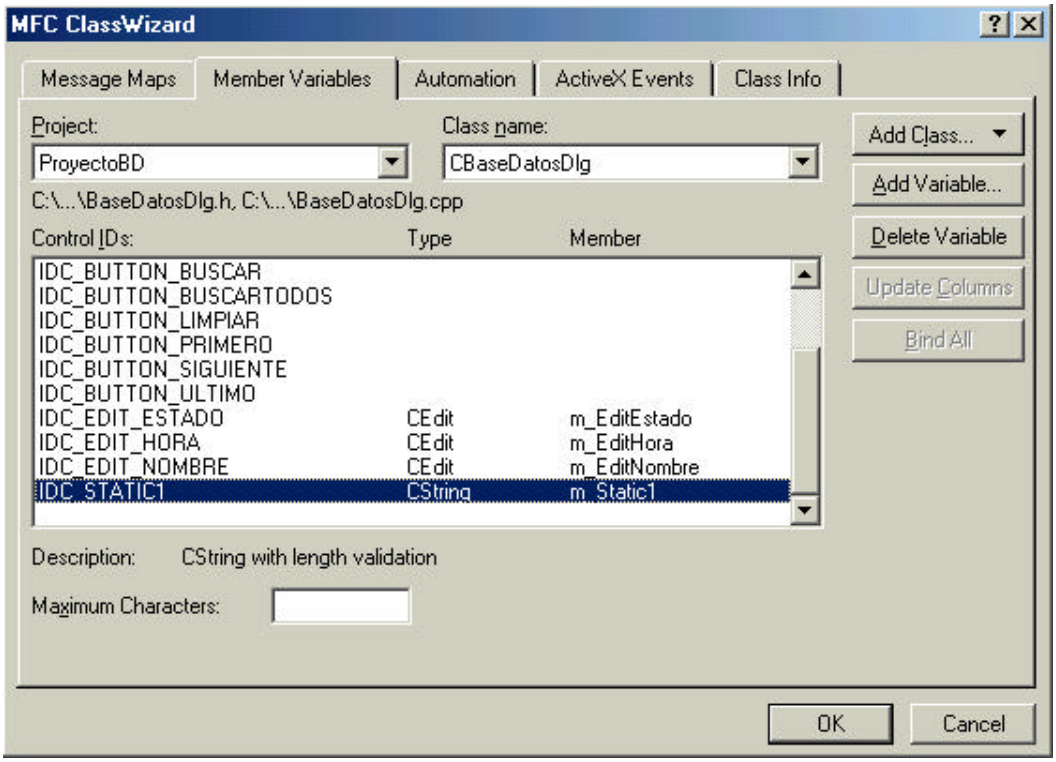
-Borrar botones Ok y Cancel.
-Insertar 3 Edit Box, 1 Static text, 10 botones.

IDC_BUTTON_PRIMERO	*Primero*
IDC_BUTTON_ANTERIOR	*Anterior*
IDC_BUTTON_ SIGUIENTE	*Siguiente*
IDC_BUTTON_ ULTIMO	*Ultimo*
IDC_BUTTON_ ANADIR	*Insertar*
IDC_BUTTON_ BORRAR	*Borrar*
IDC_BUTTON_ ACTUALIZAR	*Actualizar*
IDC_BUTTON_ BUSCARTODOS	*BuscarTodos*
IDC_BUTTON_ LIMPIAR	*Limpiar*
IDC_BUTTON_ BUSCAR	*Buscar*

-Crar variables (classwizard, member variables)

IDC_STATIC1	*m_Static1*	*CString*
IDC_EDIT_NOMBRE	*m_EditNombre*	*CEdit Control*
IDC_EDIT_ESTADO	*m_EditEstado*	*CEdit Control*
IDC_EDIT_HORA	*m_EditHora*	*CEdit Control*




MFC ClassWizard
Message Maps
Member Variables
Automation
ActiveX Events
Class Info
Project:
ProyectoBD
Class name:
CBaseDatosDlg
C:\...\BaseDatosDlg.h, C:\...\BaseDatosDlg.cpp
Control IDs:
Type
Member
IDC_BUTTON_BUSCAR
IDC_BUTTON_BUSCARTODOS
IDC_BUTTON_LIMPIAR
IDC_BUTTON_PRIMERO
IDC_BUTTON_SIGUIENTE
IDC_BUTTON_ULTIMO
IDC_EDIT_ESTADO CEdit m_EditEstado
IDC_EDIT_HORA CEdit m_EditHora
IDC_EDIT_NOMBRE CEdit m_EditNombre
IDC_STATIC1 CString m_Static1
Add Class...
Add Variable...
Delete Variable
Update Columns
Bind All
Description: CString with length validation
Maximum Characters:
OK
Cancel

Preparar la clase CBaseDatosDlg para utilizar funciones propias

-Añadir a *BaseDatosDlg.h* las declaraciones de las funciones del fichero *ABRCBaseDatosDlgUtils.h* útiles para el diálogo de base de datos.

```
class CBaseDatosDlg : public CDialog
{
// Construction
public:
        // Inicio de las declaraciones
        CString InsertarHora();
        void BorrarCampos();
        void EscribirRegistro ();
        CString Leer(CEdit *Edit);
        void Escribir(CEdit* Edit, CString String);
        //Fin de las declaraciones

        CBaseDatosSet m_BaseDatosDlgSet;
        CBaseDatosSet* m_pSet;

        CBaseDatosDlg(CWnd* pParent = NULL);   // standard constructor
```

-Añadir a *BaseDatosDlg.cpp*

#include "ABRCBaseDatosDlgUtils.h"

```
// BaseDatosDlg.cpp : implementation file
//

#include "stdafx.h"
#include "ProyectoBD.h"
#include "BaseDatosDlg.h"

#include "ABRCBaseDatosDlgUtils.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif
```

Debe estar copiado el fichero *ABRCBaseDatosDlgUtils.h* en la carpeta del proyecto, fichero que incluye las funciones útiles para el diálogo de la base de datos

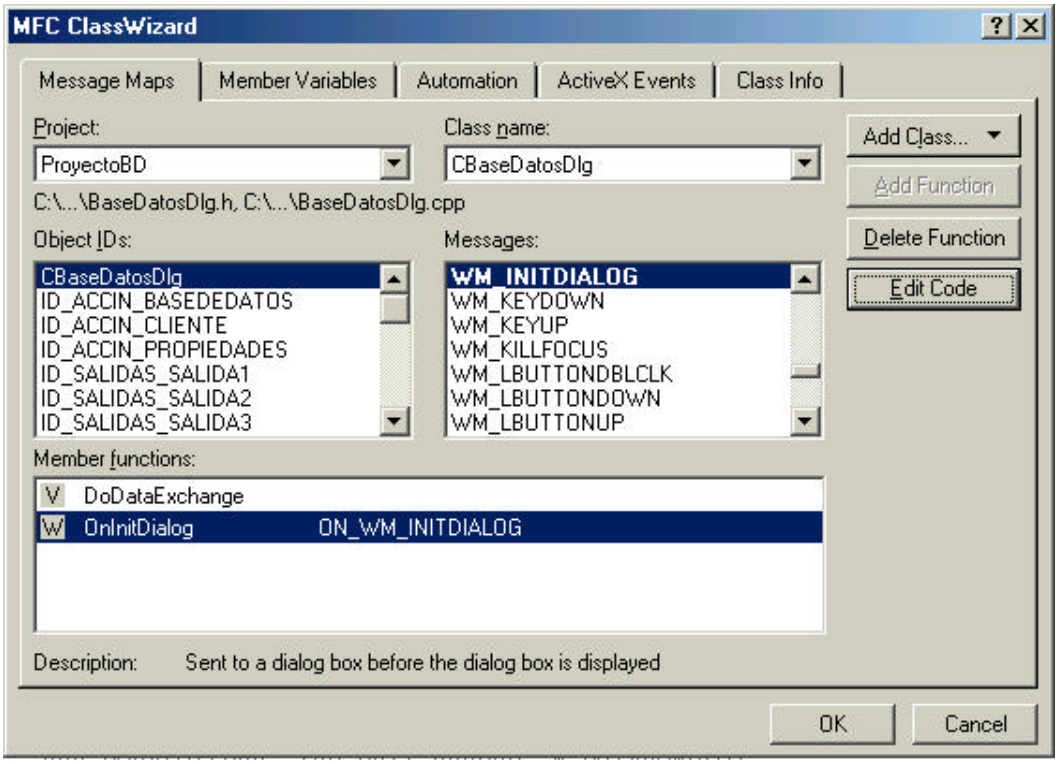
Project->Add to porject->Files->*ABRCBaseDatosDlgUtils.h*

-En *BaseDatosDlg.cpp*

-Añadir función *OnInitDialog();*

(Classwizard -> MessageMap)

(CBaseDatosDlg -> WM_INITDIALOG -> Add Function ->EditCode)

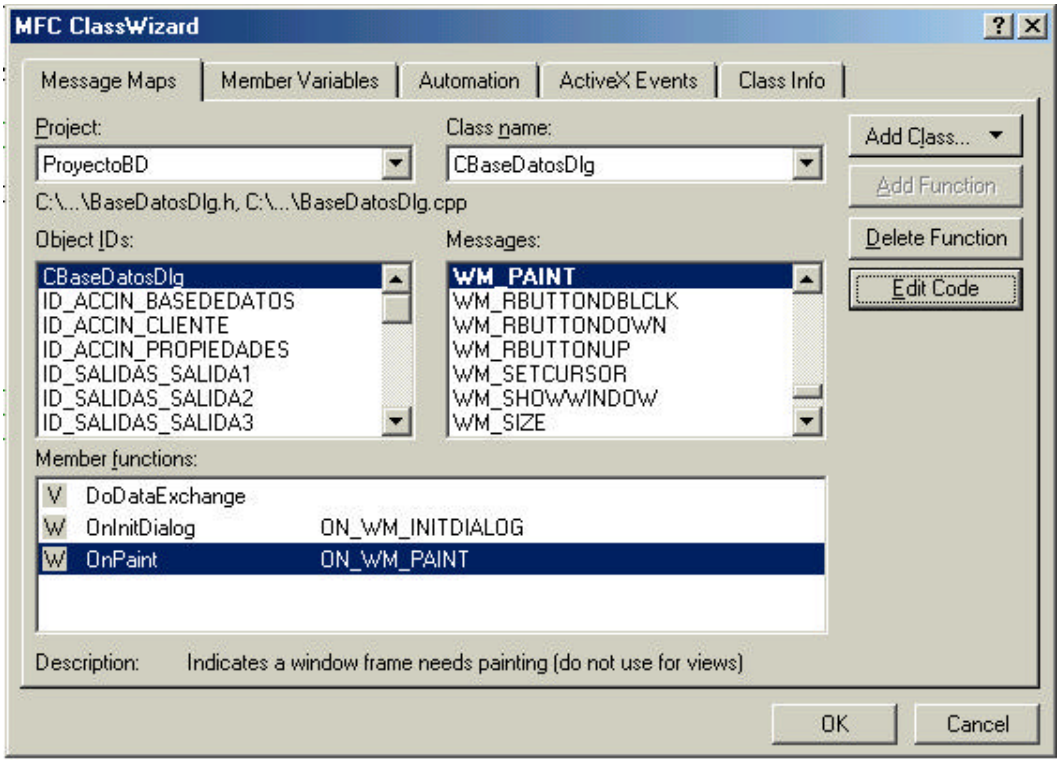



```
BOOL CBaseDatosDlg::OnInitDialog()
{
	CDialog::OnInitDialog();

	// TODO: Add extra initialization here

	m_pSet=&m_BaseDatosDlgSet ;
	m_pSet->ABRCerrarBD ();
	m_pSet->ABRAbrirBD ();
//	m_Static1=m_pSet->ABRContarRegistros();
//	UpdateData(FALSE);
//	m_pSet->ABRUltimoRegistro ();
	OnButtonBuscartodos();
	return TRUE;  // return TRUE unless you set the focus to a control
		// EXCEPTION: OCX Property Pages should return FALSE
}
```

-Añadir función *OnPaint();*

(Classwizard -> MessageMap)

(CBaseDatosDlg -> WM_PAINT -> Add Function ->EditCode)




```
void CBas eDatosDlg::OnPaint()
{
	CPaintDC dc(this); // device context for painting

	// TODO: Add your message handler code here

//	m_pSet=&m_BaseDatosDlgSet ;
//	m_pSet->ABRCerrarBD ();
//	m_pSet->ABRAbrirBD ();

	EscribirRegistro();
//	OnButtonBuscartodos();
	m_EditNombre.SetSel(m_EditNombre.GetWindowTextLength (),m_EditNombre.GetWindowTextLength (),TRUE);
	// Do not call CDialog::OnPaint() for painting messages
}
```

-Añadir las funciones

OnButtonActualizar()
(Classwizard -> MessageMap)
(CBaseDatosDlg -> IDC_BUTTON_ACTUALIZAR -> BN_CLICKED -> Add Function ->EditCode)

OnButtonAnterior()
(Classwizard -> MessageMap)
(CBaseDatosDlg -> IDC_BUTTON_ANTERIOR -> BN_CLICKED -> Add Function ->EditCode)

OnButtonBorrar()
(Classwizard -> MessageMap)
(CBaseDatosDlg -> IDC_BUTTON_BORRAR -> BN_CLICKED -> Add Function ->EditCode)

OnButtonBuscar()
(Classwizard -> MessageMap)
(CBaseDatosDlg -> IDC_BUTTON_BUSCAR -> BN_CLICKED -> Add Function ->EditCode)

OnButtonBuscarTodos()
(Classwizard -> MessageMap)
(CBaseDatosDlg -> IDC_BUTTON_BUSCARTODOS -> BN_CLICKED -> Add Function ->EditCode)

OnButtonInsertar()
(Classwizard -> MessageMap)
(CBaseDatosDlg -> IDC_BUTTON_ANADIR -> BN_CLICKED -> Add Function ->EditCode)

OnButtonLimpiar()
(Classwizard -> MessageMap)
(CBaseDatosDlg -> IDC_BUTTON_LIMPIAR -> BN_CLICKED -> Add Function ->EditCode)

OnButtonPrimero()
(Classwizard -> MessageMap)
(CBaseDatosDlg -> IDC_BUTTON_PRIMERO -> BN_CLICKED -> Add Function ->EditCode)

OnButtonSiguiente()
(Classwizard -> MessageMap)
(CBaseDatosDlg -> IDC_BUTTON_SIGUIENTE -> BN_CLICKED -> Add Function ->EditCode)

OnButtonUltimo()
(Classwizard -> MessageMap)
(CBaseDatosDlg -> IDC_BUTTON_ULTIMO -> BN_CLICKED -> Add Function ->EditCode)

Copiar el código de las funciones anteriores en *BaseDatosDlg.cpp*

```
void CBaseDatosDlg::OnButtonPrimero()
{
	// TODO: Add your control notification handler code here
	m_pSet->ABRPrimerRegistro ();
//	BorrarCampos();
	EscribirRegistro ();
}

void CBaseDatosDlg::OnButtonAnterior()
{
	// TODO: Add your control notification handler code here
	m_pSet->ABRAnteriorRegistro();
//	BorrarCampos();
	EscribirRegistro ();
}

void CBaseDatosDlg::OnButtonSiguiente()
{
	// TODO: Add your control notification handler code here
```

```
		m_pSet->ABRSiguienteRegistro();
//		OnButtonLimpiar();
		EscribirRegistro ();

}

void CBaseDatosDlg::OnButtonUltimo()
{
		// TODO: Add your control notification handler code here

		m_pSet->ABRUltimoRegistro();
//		OnButtonLimpiar();
		EscribirRegistro ();

}

void CBaseDatosDlg::OnButtonInsertar()
{
		// TODO: Add your control notification handler code here
		CString aux;


		if (Leer(&m_EditNombre)=="") m_pSet ->m_Nombre="-";
		if (Leer(&m_EditEstado)=="") m_pSet->m_Estado="-";
		m_pSet->m_Hora=InsertarHora();
		EscribirRegistro();

		m_pSet->ABRInsertarRegistro(Leer(&m_EditNombre),Leer(&m_EditEstado),Leer(&m_EditHora));

//		m_Static1=m_pSet->ABRContarRegistros();
//		UpdateData(FALSE);

		OnButtonBuscartodos();
}



void CBaseDatosDlg::OnButtonBorrar()
{
		// TODO: Add your control notification handler code here
		m_pSet->ABRBorrarRegistro (Leer(&m_EditNombre),Leer(&m_EditEstado),Leer(&m_EditHora));

		OnButtonBuscartodos();

		if (m_pSet ->GetRecordCount ()==0)
		{
			m_pSet->ABRInsertarRegistro ("Inicio de la Base de Datos","-",InsertarHora());
		}

		OnButtonBuscartodos();
		SetFocus();
}

void CBaseDatosDlg::OnButtonActualizar()
{
		// TODO: Add your control notification handler code here
		if((Leer(&m_EditNombre)!="")&&(Leer(&m_EditEstado)!="")&&(Leer(&m_EditHora)!=""))
			m_pSet->ABRActualizarRegistro (Leer(&m_EditNombre),Leer(&m_EditEstado),Leer(&m_EditHora));
}

void CBaseDatosDlg::OnButtonBuscartodos()
{
		// TODO: Add your control notification handler code here
		m_Static1=m_pSet->ABRContarRegistros();
		UpdateData(FALSE);
		m_pSet->m_Nombre ="";
		m_pSet->m_Estado ="";
		m_pSet->m_Hora ="";

		m_pSet->ABRBuscarTodosLosRegistros ();
//		OnButtonUltimo();
		EscribirRegistro ();
}

void CBaseDatosDlg::OnButtonLimpiar()
{
```

```
	// TODO: Add your control notification handler code here
	m_EditNombre.SetWindowText ("");
	m_EditEstado.SetWindowText ("");
	m_EditHora.SetWindowText ("");
//	m_pSet->m_Nombre="";
//	m_pSet->m_Estado="";
//	m_pSet->m_Hora="";
}

void CBaseDatosDlg::OnButtonBuscar()
{
	// TODO: Add your control notification handler code here
	m_pSet->m_Nombre="";
	m_pSet->m_Estado="";
	m_pSet->m_Hora="";

	m_pSet->ABRBuscarRegistro (Leer(&m_EditNombre),Leer(&m_EditEstado),Leer(&m_EditHora));

	OnButtonUltimo();

}
```

Preparar CProyectoBDDlg para utilizar el RecordSet

-Añadir a *ProyectoBDDlg.h*

#include "BaseDatosSet.h"
class CBaseDatosSet;

```
// ProyectoBDDlg.h : header file
//

#if !defined(AFX_PROYECTOBDDLG_H__14710EF7_3ED3_43D7_BACE_258C8EA5D161__INCLUDED_)
#define AFX_PROYECTOBDDLG_H__14710EF7_3ED3_43D7_BACE_258C8EA5D161__INCLUDED_

#include "BaseDatosSet.h"

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

/////////////////////////////////////////////////////////////////////////////
// CProyectoBDDlg dialog
class CBaseDatosSet;
class CBaseDatosDlg;
class CProyectoBDDlg : public CDialog
{
// Construction
```

-Insertar en la cabecera del fichero

#include "BaseDatosDlg.h"

```
#if !defined(AFX_PROYECTOBDDLG_H__14710EF7_3ED3_43D7_BACE_258C8EA5D161__INCLUDED_)
#define AFX_PROYECTOBDDLG_H__14710EF7_3ED3_43D7_BACE_258C8EA5D161__INCLUDED_

#include "BaseDatosSet.h"
#include "BaseDatosDlg.h"     // Added by ClassView

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

/////////////////////////////////////////////////////////////////////////////
// CProyectoBDDlg dialog
class CBaseDatosSet;
class CBaseDatosDlg;
```

-Crear dos variables

CBaseDatosSet m_pSet;*
CBaseDatosSet m_BaseDatosDlgSet;

```
public:
        CProyectoBDDlg(CWnd* pParent = NULL);           // standard constructor

        CBaseDatosSet* m_pSet;
        CBaseDatosSet m_BaseDatosDlgSet;

        CBaseDatosDlg* m_pBaseDatosDlg;
// Dialog Data
```

Realizar CTRL+F5 para compilar hasta aquí el proyecto

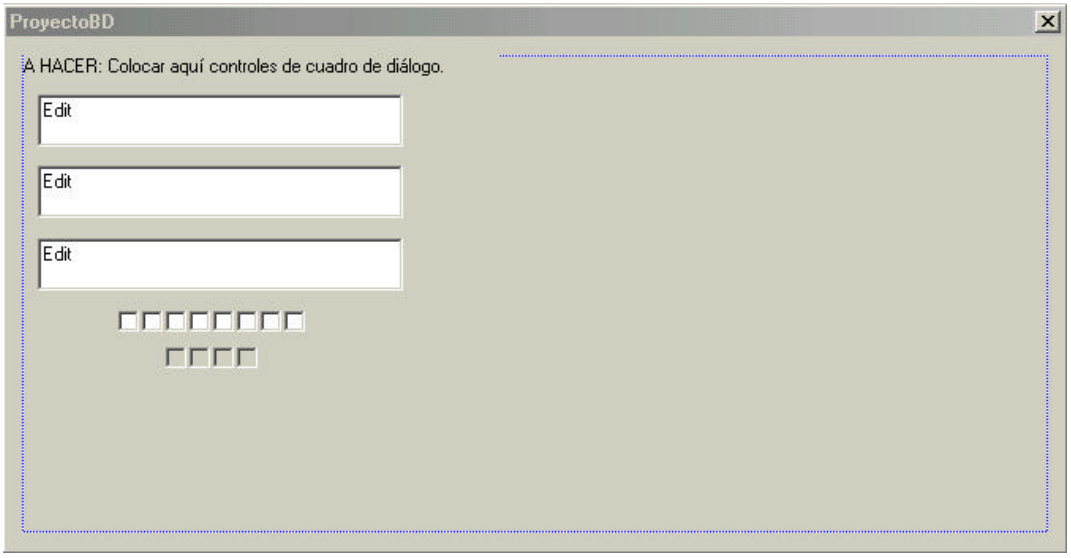
Completar diálogo principal del Proyecto
(IDD_PROYECTOBD_DIALOG)

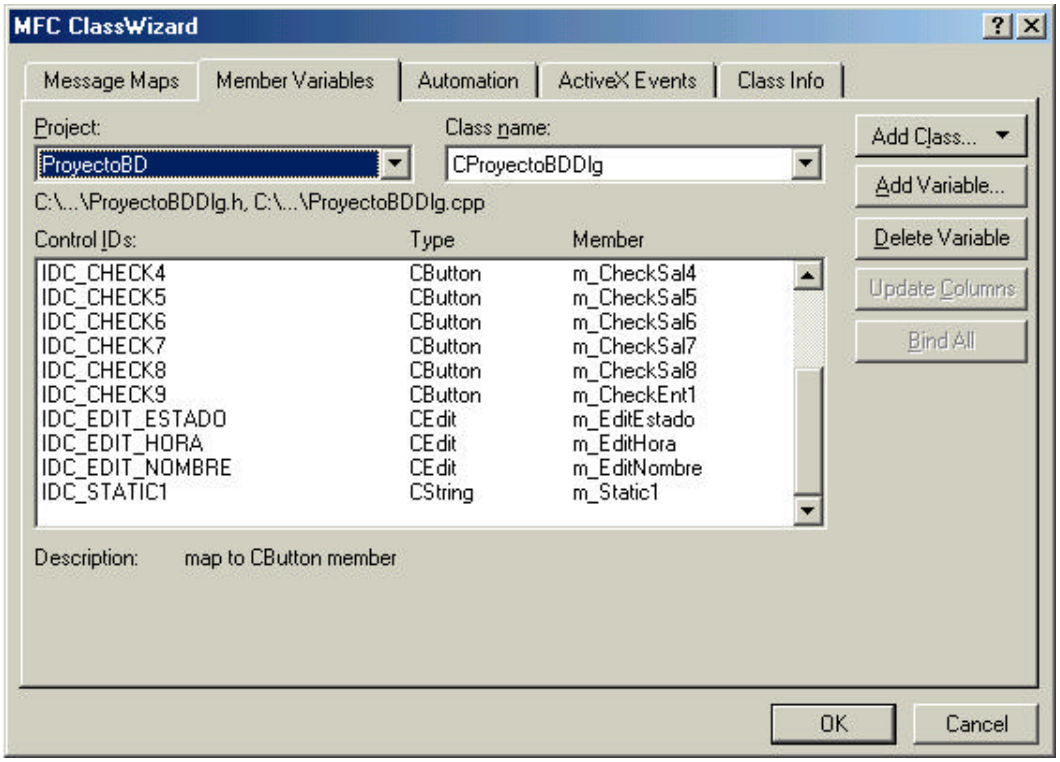
-Borrar botones aceptar y cancelar.

-Insertar 3 Edit Box, 1StaticText, 8 CheckBox, 4CheckBox (Disabled).

-Crear variables (classwizard, member variables).

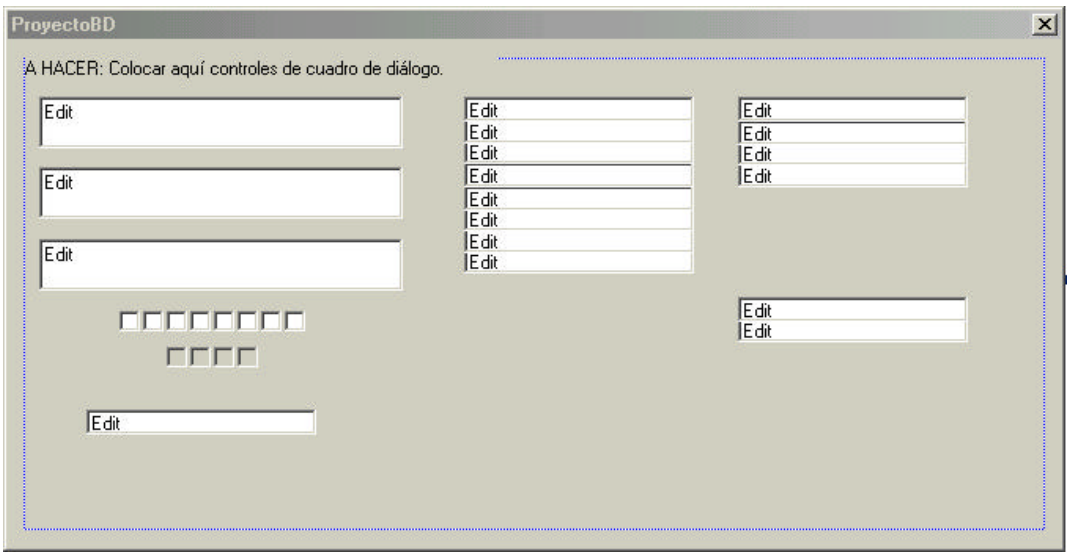
IDC_STATIC1	*m_Static1*	*CString*
IDC_EDIT_NOMBRE	*m_EditNombre*	*CEdit*
IDC_EDIT_ESTADO	*m_EditEstado*	*CEdit*
IDC_EDIT_HORA	*m_EditHora*	*CEdit*
IDC_CHECK1	*m_CheckSal1*	*CButton*
"	"	"
IDC_CHECK8	*m_CheckSal8*	*CButton*
IDC_CHECK9	*m_CheckEnt1*	*CButton*
"	"	"
IDC_CHECK12	*m_CheckEnt4*	*CButton*





-En el diálogo principal en CProyectoBDDlg (*IDD_PROYECTOBD_DIALOG*)

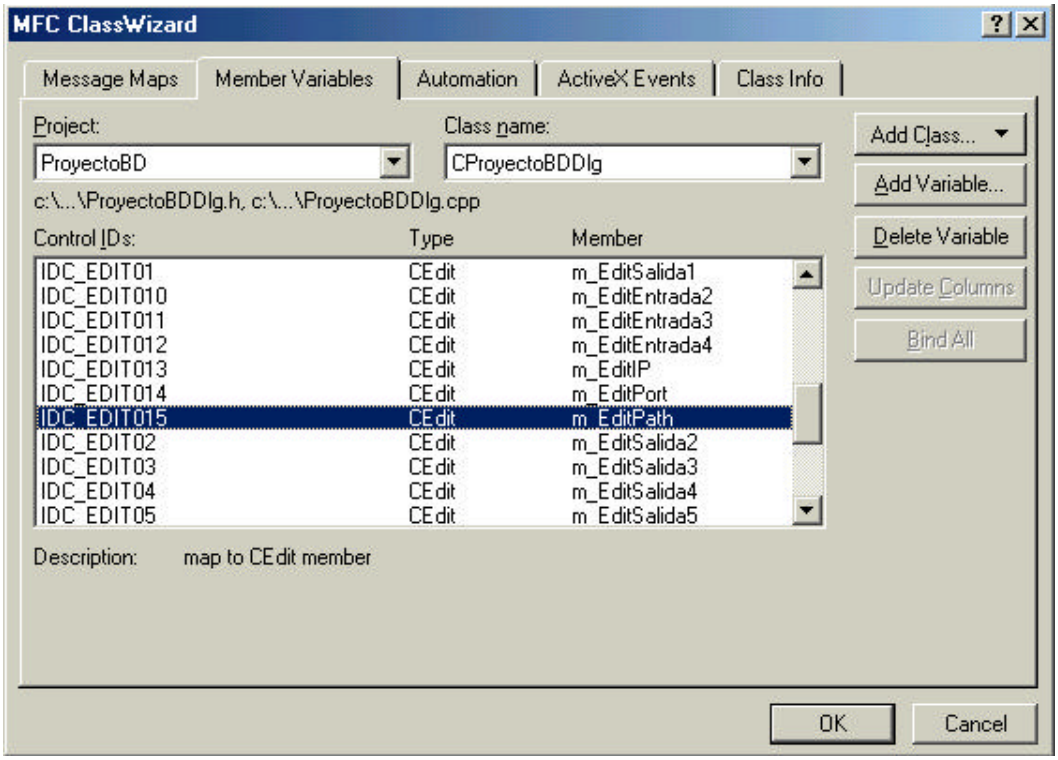
Añadir 15 EditBox.



-Crear variables (Cla ssWizard -> Member variables)

IDC_EDIT01	*CEdit*	*m_EditSalida1*
"	*"*	*"*
IDC_EDIT08	*CEdit*	*m_EditSalida8*

IDC_EDIT09	*CEdit*	*m_EditEntrada1*
"	"	"
IDC_EDIT012	*CEdit*	*m_EditEntrada4*
IDC_EDIT013	*CEdit*	*m_EditIP*
IDC_EDIT014	*CEdit*	*m_EditPort*
IDC_EDIT015	*CEdit*	*m_EditPath*



-En *BaseDatosSet.cpp*

Modificar función *GetDefaultConnect()*

```
CString aux;

if (AfxGetApp()->GetProfileString("BD","Path")=="")
{
AfxGetApp()->WriteProfileString("BD","Path","bd1.mdb");
AfxGetApp()->WriteProfileString("Socket","Port","888");
AfxGetApp()->WriteProfileString("Leyenda","Salida1","Salida1");
                                "
AfxGetApp()->WriteProfileString("Socket","IP","127.0.0.1");
}

aux="ODBC;DRIVER={MICROSOFT ACCESS DRIVER (*.mdb)};DSN='';DBQ='';
aux.Insert(aux.GetLength();AfxGetApp()->GetProfileString("BD","Path");
return _T(aux);
```

```
CString CBaseDatosSet::GetDefaultConnect()
{
//      return _T("ODBC;DRIVER={MICROSOFT ACCESS DRIVER
(*.mdb)};DSN='';DBQ=C:\\windows\\escritorio\\stdreg32.mdb");

        CString aux;
        if (AfxGetApp()->GetProfileString ("BD","Path")=="")
        {
                AfxGetApp()->WriteProfileString ("Socket","Port","888");
                AfxGetApp()->WriteProfileString ("BD","Path","bd1.mdb");

                AfxGetApp()->WriteProfileString ("Leyenda","Salida1","Salida1");
                AfxGetApp()->WriteProfileString ("Leyenda","Salida2","Salida2");
                AfxGetApp()->WriteProfileString ("Leyenda","Salida3","Salida3");
                AfxGetApp()->WriteProfileString ("Leyenda","Salida4","Salida4");
                AfxGetApp()->WriteProfileString ("Leyenda","Salida5","Salida5");
                AfxGetApp()->WriteProfileString ("Leyenda","Salida6","Salida6");
                AfxGetApp()->WriteProfileString ("Leyenda","Salida7","Salida7");
                AfxGetApp()->WriteProfileString ("Leyenda","Salida8","Salida8");

                AfxGetApp()->WriteProfileString ("Leyenda","Entrada1","Entrada1");
                AfxGetApp()->WriteProfileString ("Leyenda","Entrada2","Entrada2");
                AfxGetApp()->WriteProfileString ("Leyenda","Entrada3","Entrada3");
                AfxGetApp()->WriteProfileString ("Leyenda","Entrada4","Entrada4");

                AfxGetApp()->WriteProfileString ("Socket","IP","127.0.0.1");
        }
        aux="ODBC;DRIVER={MICROSOFT ACCESS DRIVER (*.mdb)};DSN='';DBQ=";
        aux.Insert (aux.GetLength (),AfxGetApp()->GetProfileString ("BD","Path"));
        return _T(aux);

//      return _T("ODBC;DRIVER={MICROSOFT ACCESS DRIVER
(*.mdb)};DSN='';DBQ=C:\\windows\\escritorio\\bd1.mdb");
//      En el path de la base de datos de arriba es necesario poner dos barras debido a
estar entre comillas

}
```

-En *ProyectoBD.cpp*

-Insertar en la función *InitInstance()*

Después de *Enable3dControls_Static()*

#endif

SetRegistryKey("ABRProyectos");

```
BOOL CProyectoBDApp::InitInstance()
{
        AfxEnableControlContainer();

        // Standard initialization
        // If you are not using these features and wish to reduce the size
        //  of your final executable, you should remove from the following
        //  the specific initialization routines you do not need.

#ifdef _AFXDLL
        Enable3dControls();                     // Call this when using MFC in a shared DLL
#else
        Enable3dControlsStatic();     // Call this when linking to MFC statically
#endif
        SetRegistryKey("ABRProyectos"); //Entradas al Registro ABR

        CProyectoBDDlg dlg;
        m_pMainWnd = &dlg;
        int nResponse = dlg.DoModal();
        if (nResponse == IDOK)
        {
                // TODO: Place code here to handle when the dialog is
                //  dismissed with OK
        }
        else if (nResponse == IDCANCEL)
        {
                // TODO: Place code here to handle when the dialog is
                //  dismissed with Cancel
```

```
        }

        // Since the dialog has been closed, return FALSE so that we exit the
        //  application, rather than start the application's message pump.
        return FALSE;
}
```

-Crear variables en *Proyecto BDDlg.h*

int Salida1; int Salida2; int Salida3; int Salida4; int Salida5;
int Salida6; int Salida7; int Salida8; int Entrada1; int Entrada2;
int Entrada3; int Entrada4;

```
// Construction
public:
        CProyectoBDDlg(CWnd* pParent = NULL);        // standard constructor

        int Salida1;
        int Salida2;
        int Salida3;
        int Salida4;
        int Salida5;
        int Salida6;
        int Salida7;
        int Salida8;
        int Entrada1;
        int Entrada2;
        int Entrada3;
        int Entrada4;

        CBaseDatosSet* m_pSet;
        CBaseDatosSet m_BaseDatosDlgSet;

        CBaseDatosDlg* m_pBaseDatosDlg;
```

Preparar la clase CProyectoBDDlg para utilizar funciones propias

-Añadir a *ProyectoBDDlg.h* las declaraciones de las funciones de *"ABRCProyectoBDDlgUtils.h"* útiles para el diálogo principal del proyecto

```
class CProyectoBDDlg : public CDialog
{
// Construction
public:
	CProyectoBDDlg(CWnd* pParent = NULL);		// standard constructor
	// incicio de las declaraciones

	void EnviarLeyendaAlSck();
	void ImprimirLeyenda();
	int m_nTimer1;
	void LeerEntradasPuerto();
	void EnviarEstadoAlSck();
	void GuardarSalidas();
	CString Leer(CEdit *Edit);
	CString MirarEstado(int nSal);
	void InicializarSalidas();
	void InsertarRegistro(CString Nombre,CString Estado, CString Hora);
	CString InsertarHora();
	void EscribirRegistro ();

	void ActualizarPuerto();


	// fin de las declaraciones

	int Salida1;
	int Salida2;
```

-Añadir a *ProyectoBDDlg.cpp*

#include "ABRCProyectoBDDlgUtils.h"

```
// ProyectoBDDlg.cpp : implementation file
//

#include "stdafx.h"
#include "ProyectoBD.h"
#include "ProyectoBDDlg.h"

#include "BaseDatosDlg.h"
#include "ABRCProyectoBDDlgUtils.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif
```

Debe estar copiado el fichero *ABRCProyectoBDDlgUtils.h* en la carpeta del proyecto, fichero que incluye las funciones útiles para el diálogo principal de la aplicación.

Project->Add to project -> Files -> *ABRCProyectoBDDlgUtils.h*

En *ProyectoBDDlg.cpp*

-Modificar la función *OnInitDialog()*

(Classwizard -> MessageMap)
(CProyectoBDDlg -> WM_INITDIALOG ->EditCode)

```
BOOL CProyectoBDDlg::OnInitDialog()
{
	CDialog::OnInitDialog();
```

```
	// Add "About..." menu item to system menu.

	// IDM_ABOUTBOX must be in the system command range.
	ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
	ASSERT(IDM_ABOUTBOX < 0xF000);

	CMenu* pSysMenu = GetSystemMenu(FALSE);
	if (pSysMenu != NULL)
	{
		CString strAboutMenu;
		strAboutMenu.LoadString(IDS_ABOUTBOX);
		if (!strAboutMenu.IsEmpty())
		{
			pSysMenu->AppendMenu(MF_SEPARATOR);
			pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
		}
	}

	// Set the icon for this dialog.  The framework does this automatically
	//  when the application's main window is not a dialog
	SetIcon(m_hIcon, TRUE);			// Set big icon
	SetIcon(m_hIcon, FALSE);		// Set small icon

	// TODO: Add extra initialization here

	m_nTimer1=SetTimer(1,500,NULL);

	m_pSet=&m_BaseDatosDlgSet;
	m_pSet->ABRAbrirBD ();

	InicializarSalidas();
	m_pSet->ABRCerrarBD ();
	ImprimirLeyenda();

	m_Socket1.SetLocalPort (atoi(Leer(&m_EditPort)));
	m_Socket1.ABREscucharSck ();
	m_Static1=m_Socket1.ABRConvertirEstado (m_Socket1.GetState ());
	UpdateData(FALSE);

	return TRUE;  // return TRUE  unless you set the focus to a control
}
```

-Modificar la función *OnPaint()*
(Classwizard -> MessageMap)
(CProyectoBDDlg -> WM_PAINT ->EditCode)

```
void CProyectoBDDlg::OnPaint()
{
	if (IsIconic())
	{
		CPaintDC dc(this); // device context for painting

		SendMessage(WM_ICONERASEBKGND, (WPARAM) dc.GetSafeHdc(), 0);

		// Center icon in client rectangle
		int cxIcon = GetSystemMetrics(SM_CXICON);
		int cyIcon = GetSystemMetrics(SM_CYICON);
		CRect rect;
		GetClientRect(&rect);
		int x = (rect.Width() - cxIcon + 1) / 2;
		int y = (rect.Height() - cyIcon + 1) / 2;

		// Draw the icon
		dc.DrawIcon(x, y, m_hIcon);
	}
	else
	{
		CDialog::OnPaint();
	}
	m_EditNombre.SetSel(m_EditNombre.GetWindowTextLength (),m_EditNombre.GetWindowTextLength (),TRUE);
}
```

-Añadir la función *OnDestroy()*
(Classwizard -> MessageMap)
(CProyectoBDDlg -> WM_DESTROY ->Add Function->EditCode)

```
void CProyectoBDDlg::OnDestroy()
{
	CDialog::OnDestroy();

	// TODO: Add your message handler code here
	m_pSet=&m_BaseDatosDlgSet;
	m_pSet->ABRAbrirBD ();

	GuardarSalidas();

	m_pSet->ABRCerrarBD ();

}
```

-Añadir las funciones *OnSalidasSalida1() ... OnSalidasSalida8()*

OnSalidasSalida1()
(Classwizard -> MessageMap)
(CProyectoBDDlg -> ID_SALIDAS_SALIDA1-> COMMAND -> Add Function ->EditCode)

OnSalidasSalida2()
(Classwizard -> MessageMap)
(CProyectoBDDlg -> ID_SALIDAS_SALIDA2-> COMMAND -> Add Function ->EditCode)

OnSalidasSalida3()
(Classwizard -> MessageMap)
(CProyectoBDDlg -> ID_SALIDAS_SALIDA3-> COMMAND -> Add Function ->EditCode)

OnSalidasSalida4()
(Classwizard -> MessageMap)
(CProyectoBDDlg -> ID_SALIDAS_SALIDA4-> COMMAND -> Add Function ->EditCode)

OnSalidasSalida5()
(Classwizard -> MessageMap)
(CProyectoBDDlg -> ID_SALIDAS_SALIDA5-> COMMAND -> Add Function ->EditCode)

OnSalidasSalida6()
(Classwizard -> MessageMap)
(CProyectoBDDlg -> ID_SALIDAS_SALIDA6-> COMMAND -> Add Function ->EditCode)

OnSalidasSalida7()
(Classwizard -> MessageMap)
(CProyectoBDDlg -> ID_SALIDAS_SALIDA7-> COMMAND -> Add Function ->EditCode)

OnSalidasSalida8()
(Classwizard -> MessageMap)
(CProyectoBDDlg -> ID_SALIDAS_SALIDA8-> COMMAND -> Add Function ->EditCode)

Copiar el código de las funciones anteriores *ProyectoBDDlg.cpp*

```
void CProyectoBDDlg::OnSalidasSalida1()
{
	// TODO: Add your command handler code here
	InsertarRegistro(Leer(&m_EditSalida1),MirarEstado(1),InsertarHora());
	ActualizarPuerto();
}

void CProyectoBDDlg::OnSalidasSalida2()
{
	// TODO: Add your command handler code here
		InsertarRegistro(Leer(&m_EditSalida2),MirarEstado(2),InsertarHora());
		ActualizarPuerto();
}

void CProyectoBDDlg::OnSalidasSalida3()
{
	// TODO: Add your command handler code here
		InsertarRegistro(Leer(&m_EditSalida3),MirarEstado(3),InsertarHora());
		ActualizarPuerto();
}

void CProyectoBDDlg::OnSalidasSalida4()
{
	// TODO: Add your command handler code here
		InsertarRegistro(Leer(&m_EditSalida4),MirarEstado(4),InsertarHora());
		ActualizarPuerto();
}

void CProyectoBDDlg::OnSalidasSalida5()
{
	// TODO: Add your command handler code here
		InsertarRegistro(Leer(&m_EditSalida5),MirarEstado(5),InsertarHora());
		ActualizarPuerto();
}

void CProyectoBDDlg::OnSalidasSalida6()
{
	// TODO: Add your command handler code here
		InsertarRegistro(Leer(&m_EditSalida6),MirarEstado(6),InsertarHora());
		ActualizarPuerto();
}

void CProyectoBDDlg::OnSalidasSalida7()
{
	// TODO: Add your command handler code here
		InsertarRegistro(Leer(&m_EditSalida7),MirarEstado(7),InsertarHora());
		ActualizarPuerto();
}

void CProyectoBDDlg::OnSalidasSalida8()
{
	// TODO: Add your command handler code here
		InsertarRegistro(Leer(&m_EditSalida8),MirarEstado(8),InsertarHora());
		ActualizarPuerto();
}
```

-Añadir funciones *OnCheck1()* ... *OnCheck8()*

OnCheck1()
(Classwizard -> MessageMap)

(CProyectoBDDlg -> IDC_CHECK1 -> BN_CLICKED -> Add Function -
>EditCode)

OnCheck2()
(Classwizard -> MessageMap)
(CProyectoBDDlg -> IDC_CHECK2 -> BN_CLICKED -> Add Function -
>EditCode)

OnCheck3()
(Classwizard -> MessageMap)
(CProyectoBDDlg -> IDC_CHECK3 -> BN_CLICKED -> Add Function -
>EditCode)

OnCheck4()
(Classwizard -> MessageMap)
(CProyectoBDDlg -> IDC_CHECK4 -> BN_CLICKED -> Add Function -
>EditCode)

OnCheck5()
(Classwizard -> MessageMap)
(CProyectoBDDlg -> IDC_CHECK5 -> BN_CLICKED -> Add Function -
>EditCode)

OnCheck6()
(Classwizard -> MessageMap)
(CProyectoBDDlg -> IDC_CHECK6 -> BN_CLICKED -> Add Function -
>EditCode)

OnCheck7()
(Classwizard -> MessageMap)
(CProyectoBDDlg -> IDC_CHECK7 -> BN_CLICKED -> Add Function -
>EditCode)

OnCheck8()
(Classwizard -> MessageMap)
(CProyectoBDDlg -> IDC_CHECK8 -> BN_CLICKED -> Add Function -
>EditCode)

Copiar el código de las funciones anteriores *ProyectoBDDlg.cpp*

```
void CProyectoBDDlg::OnCheck1()
{
	// TODO: Add your control notification handler code here
	OnSalidasSalida1();
}

void CProyectoBDDlg::OnCheck2()
{
	// TODO: Add your control notification handler code here
	OnSalidasSalida2();
}

void CProyectoBDDlg::OnCheck3()
{
	// TODO: Add your control notification handler code here
	OnSalidasSalida3();
```

```
}

void CProyectoBDDlg::OnCheck4()
{
	// TODO: Add your control notification handler code here
	OnSalidasSalida4();
}

void CProyectoBDDlg::OnCheck5()
{
	// TODO: Add your control notification handler code here
	OnSalidasSalida5();
}

void CProyectoBDDlg::OnCheck6()
{
	// TODO: Add your control notification handler code here
	OnSalidasSalida6();
}

void CProyectoBDDlg::OnCheck7()
{
	// TODO: Add your control notification handler code here
	OnSalidasSalida7();
}

void CProyectoBDDlg::OnCheck8()
{
	// TODO: Add your control notification handler code here
	OnSalidasSalida8();
}
```

Añadir a CProyectoBDDlg la clase para controlar el puerto paralelo

-Debemos añadir la clase *PuertoParalelo.cpp* y *PuertoParalelo.h*
Project->AddtoProject->Files -> *PuertoParalelo.cpp*
Project->AddtoProject->Files -> *PuertoParalelo.h*

-En *ProyectoBDDlg.h*

-Insertar en la cabecera del fichero
#include "PuertoParalelo.h"

```
#if !defined(AFX_PROYECTOBDDLG_H__14710EF7_3ED3_43D7_BACE_258C8EA5D161__INCLUDED_)
#define AFX_PROYECTOBDDLG_H__14710EF7_3ED3_43D7_BACE_258C8EA5D161__INCLUDED_

#include "BaseDatosSet.h"
#include "PuertoParalelo.h"   // Added by ClassView
#include "BaseDatosDlg.h"     // Added by ClassView

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

/////////////////////////////////////////////////////////////////////////////
// CProyectoBDDlg dialog
class CBaseDatosSet;
class CBaseDatosDlg;
#include "Propiedades1Dlg.h"
#include "ClienteD lg.h"
```

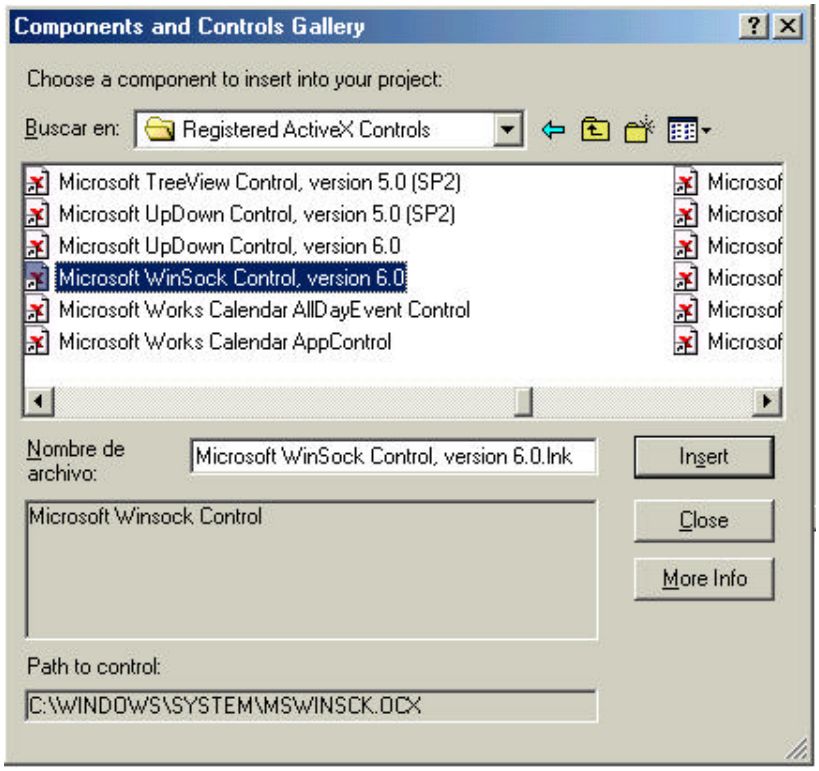
-Crear variable
CPuertoParalelo PuertoParalelo;

```
void EscribirRegistro ();

// fin de las declaraciones
CPuertoParalelo PuertoParalelo;

int Salida1;
int Salida2;
```

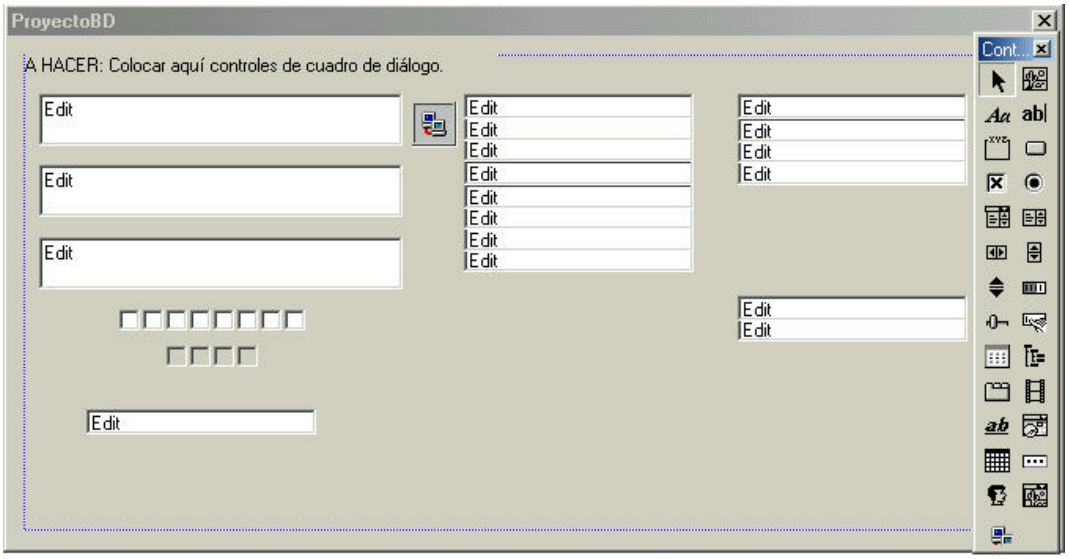
Añadir a CProyectoBDDlg funciones de servidor

-Insertar Control Winsock

Project->Add to Project -> Components and Controls-> Registered ActiveX Controls -> Microsoft Winsock Control

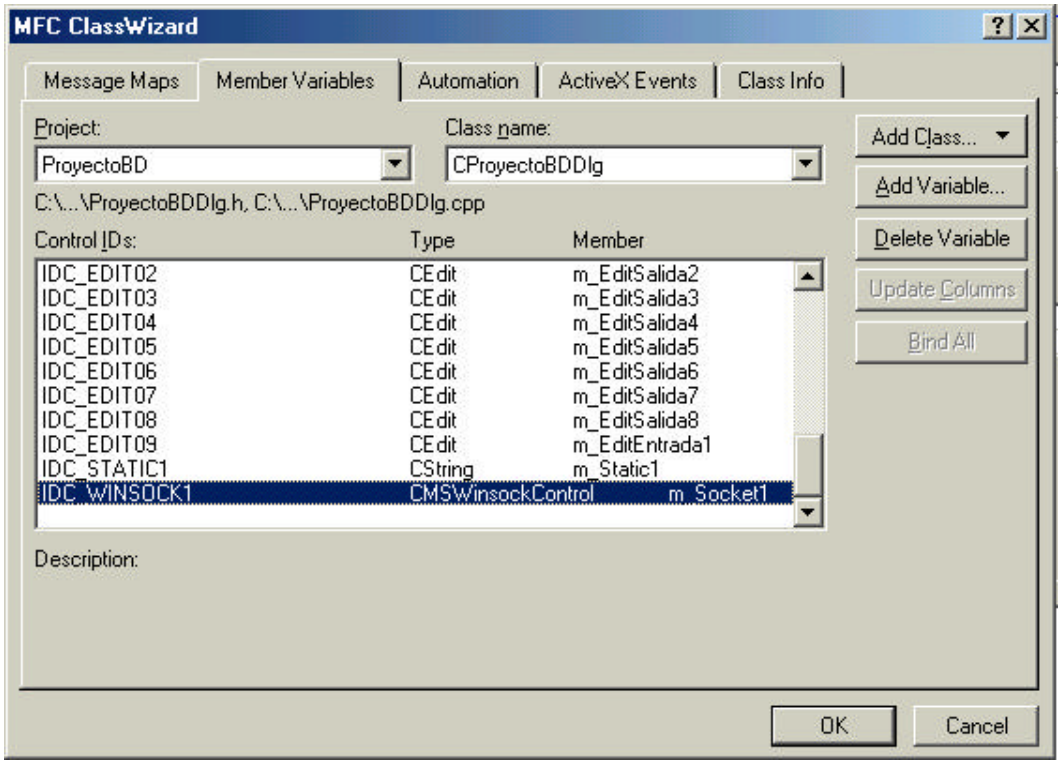


-Insertar control CMSWinsockControl al diálogo principal *(IDD_PROYECTOBD_DIALOG)*

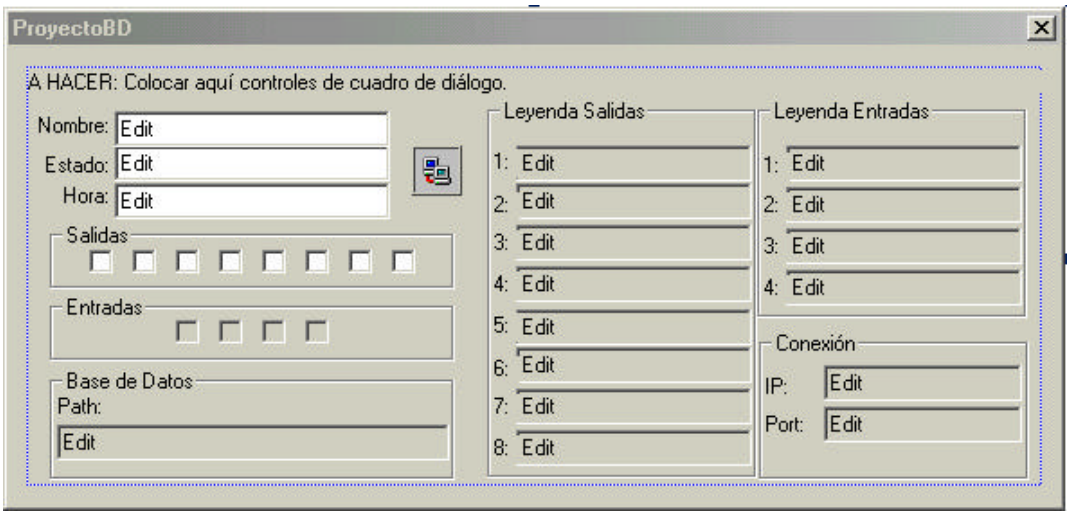


-Crear variable *m_Socket1;* (classwizard, member variables)

IDC_WINSOCK1 *m_Socket1* *Control*



-Finalización del diálogo principal*(IDD_PROYECTOBD_DIALOG)*



Preparar la clase CMSWinsockControl para utilizar funciones propias

-Añadir a *MSWinsockControl.h* las declaraciones de las funciones del fichero *ABRWinsock.h* para utilizar el control Winsock i la variable *CString Dato*.

```
// Operations
public:
        //Variable para recoger los datos recibidos con el DataArrival
       CString Dato;

       //Declaraciones de las funciones del fichero ABRWinsock.h

       void CMSWinsockControl::ABRConectarSck();
       void CMSWinsockControl::ABROnConnectWinsock1();
       void CMSWinsockControl::ABREscucharSck();
       void CMSWinsockControl::ABROnConnectionRequestWinsock1(long requestID);
       void CMSWinsockControl::ABROnDataArrivalWinsock1(long bytesTotal);
       void CMSWinsockControl::ABREnviarSck(CString Datos);
       void CMSWinsockControl::ABROnCloseWinsock1();
       void CMSWinsockControl::ABROnErrorWinsock1(short Number, BSTR FAR* Description,
long Scode, LPCTSTR Source, LPCTSTR HelpFile, long HelpContext, BOOL FAR*
CancelDisplay);
       void CMSWinsockControl::ABRCerrarSck();
       void CMSWinsockControl::ABROnSendCompleteWinsock1();
       CString CMSWinsockControl::ABRConvertirEstado(int Estado);
       //Fin de las declaraciones

       long GetProtocol();
```

-Añadir a *MSWinsockControl.cpp*

Después de *#include "mswinsockcontrol.h"*
#include "ABRWinsock.h"

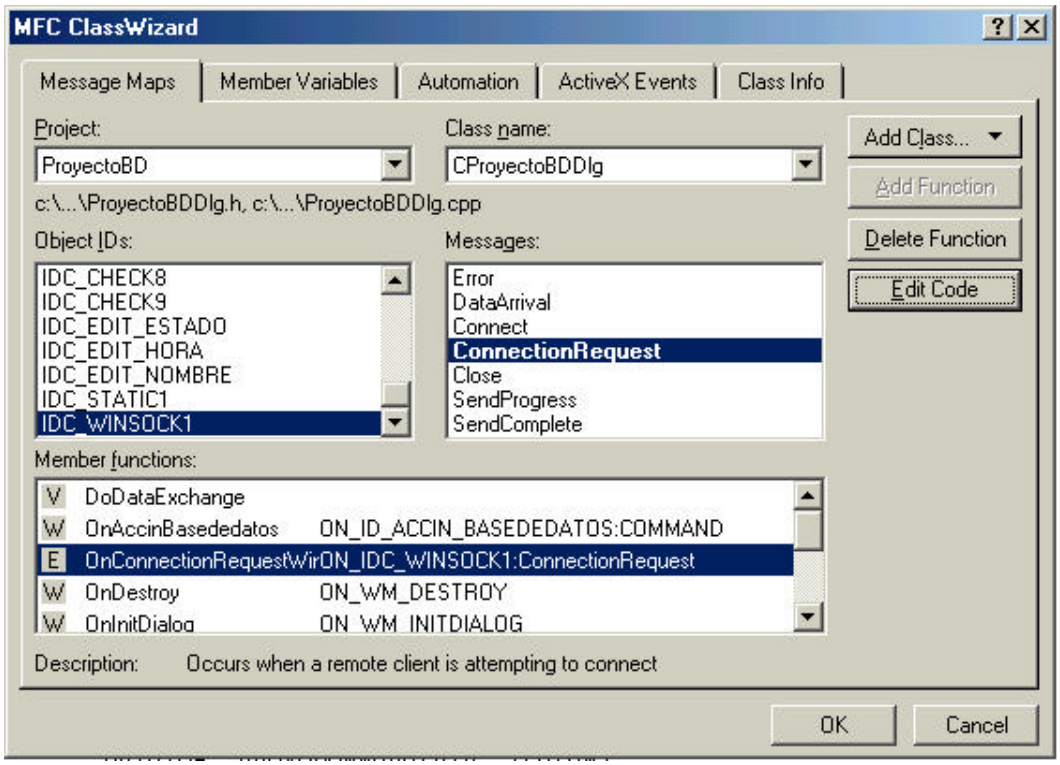
```
// Machine generated IDispatch wrapper class(es) created by Microsoft Visual C++

// NOTE: Do not modify the contents of this file.  If this class is regenerated by
//  Microsoft Visual C++, your modifications will be overwritten.


#include "stdafx.h"
#include "mswinsockcontrol.h"

#include "ABRWinsock.h"
```

-Copiar *ABRWinsock.h* en la carpeta del proyecto cliente,fichero donde se encuentran las funciones ABR para la utilización del Control Winsock.

Project->Add to Project->Files->ABRWinsock.h

-Insertar función *OnConnectionRequestWinsock1 (long request ID);*

(Classwizard -> MessageMap)
(IDC_WINSOCK1 -> ConnectionRequest ->Add Function->EditCode)



m_Socket1.ABROnConnectionRequestWinsock1(request ID);
m_Static1=m_Socket1.ABRConvertirEstado(m_Socket1.GetState());
UpdateData(FALSE);
EnviarEstadoalSck();

```
void CProyectoBDDlg::OnConnectionRequestWinsock1(long requestID)
{
	// TODO: Add your control notification handler code here
	m_Socket1.ABROnConnectionRequestWinsock1 (requestID);
	m_Static1=m_Socket1.ABRConvertirEstado (m_Socket1.GetState ());
	UpdateData(FALSE);

	//Enviar Estado Inicial
	EnviarLeyendaAlSck();
//	EnviarEstadoAlSck();
}
```

-Insertar función *OnCloseWinsock1()*
(Classwizard -> MessageMap)
(IDC_WINSOCK1 -> Close ->Add Function->EditCode)

m_Socket1.ABROnCloseWinsock1();
m_Static1=m_Socket1.ABRConvertirEstado(mSocket1.GetState());
UpdateData(FALSE);
m_Socket1.ABRCerrarSck();
m_Static1=...
UpdateDAta(FALSE);

```
void CProyectoBDDlg::OnCloseWinsock1()
{
	// TODO: Add your control notification handler code here
```

```
        m_Socket1.ABROnCloseWinsock1 ();
        m_Static1=m_Socket1.ABRConvertirEstado (m_Socket1.GetState ());
        UpdateData(FALSE);
        m_Socket1.ABRCerrarSck ();
        m_Static1=m_Socket1.ABRConvertirEstado (m_Socket1.GetState ());
        UpdateData(FALSE);
        m_Socket1.ABREscucharSck ();
        m_Static1=m_Socket1.ABRConvertirEstado (m_Socket1.GetState ());
        UpdateData(FALSE);

}
```

-Insertar función *OnDataArrivalWinsock1(long bytes total)*

(Classwizard -> MessageMap)

(IDC_WINSOCK1 -> DataArrival ->Add Function->EditCode)

m_Socket1.ABROnDataArrivalWinsock(bytes Total);
m_Static1="Recibido: "
m_Static1.Insert(m_Static1.getLength(),m_Socket1.Dato);
UpdateData(FALSE);
if(m_Socket1.Dato=="Salida1") OnSalidasSalida1();
"
if(m_Socket1.Dato=="Salida8") OnSalidasSalida8();

```
void CProyectoBDDlg::OnDataArrivalWinsock1(long bytesTotal)
{
        // TODO: Add your control notification handler code here
        m_Socket1.ABROnDataArrivalWinsock1 (bytesTotal);
        m_Static1="Recibido: ";
        m_Static1.Insert(m_Static1.GetLength (),m_Socket1.Dato );
        UpdateData(FALSE);

        if (m_Socket1.Dato=="Salida1") OnSalidasSalida1();
        if (m_Socket1.Dato=="Salida2") OnSalidasSalida2();
        if (m_Socket1.Dato=="Salida3") OnSalidasSalida3();
        if (m_Socket1.Dato=="Salida4") OnSalidasSalida4();
        if (m_Socket1.Dato=="Salida5") OnSalidasSalida5();
        if (m_Socket1.Dato=="Salida6") OnSalidasSalida6();
        if (m_Socket1.Dato=="Salida7") OnSalidasSalida7();
        if (m_Socket1.Dato=="Salida8") OnSalidasSalida8();
}
```

-Insertar función *OnTimer(UINT nIDEvent)*

(Classwizard -> MessageMap)

(CProyectoBDDlg -> WM_TIMER ->Add Function->EditCode)

LeerEntradasPuerto();

```
void CProyectoBDDlg::OnTimer(UINT nIDEvent)
{
        // TODO: Add your message handler code here and/or call default
        LeerEntradasPuerto();
        CDialog::OnTimer(nIDEvent);
}
```

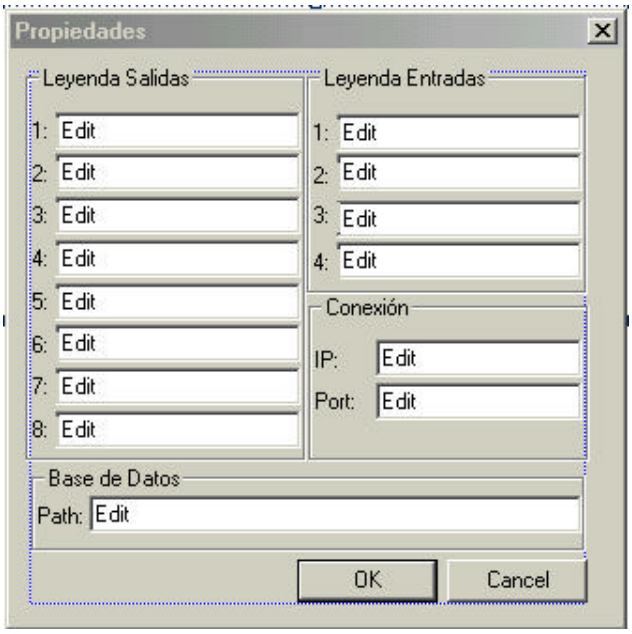
Realizar CTRL+F5 para compilar hasta aquí el proyecto

Añadir a CProyectoBDDlg un dialogo de propiedades

-Crear un diálogo para modificar las propiedades del proyecto (*IDD_PROPIEDADES1DLG_DIALOG*)

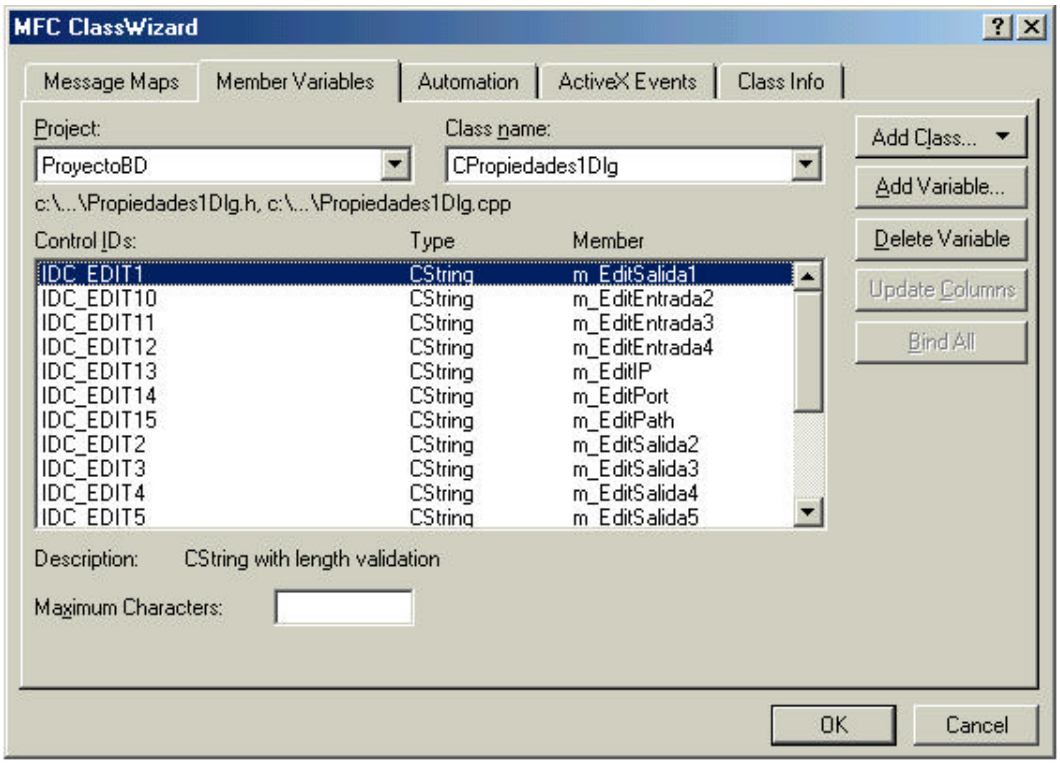
Insert->New Class -> Form Class-> *CPropiedades1Dlg*

-Insertar 15 EditBox



-Crear variables (Classwizard -> Member variables)

IDC_EDIT1	*CString*	*m_EditSalida1*
"	*"*	*"*
IDC_EDIT8	*CString*	*m_EditSalida8*
IDC_EDIT9	*CString*	*m_EditEntrada1*
"	*"*	*"*
IDC_EDIT12	*CString*	*m_EditEntrada4*
IDC_EDIT13	*CString*	*m_EditIP*
IDC_EDIT14	*CString*	*m_EditPort*
IDC_EDIT15	*CString*	*m_EditPath*



-En *Propiedades1Dlg.cpp*

-Insertar en función *OnOk()*

(Classwizard -> MessageMap)
(IDOK -> BN_CLICKED ->Add Function->EditCode)

UpdateData(TRUE);
AfxGetApp()->WriteProfileString("Leyenda","Salida1",m_EditSalida1);
"
AfxGetApp()->WriteProfileString("Leyenda","Salida8",m_EditSalida8);

AfxGetApp()->WriteProfileString("Leyenda","Entrada1",m_EditEntrada1);
"
AfxGetApp()->WriteProfileString("Leyenda","Entrada4",m_EditEntrada4);

AfxGetApp()->WriteProfileString("Socket","IP",m_EditIP);
AfxGetApp()->WriteProfileString("Socket","Port",m_EditPort);
AfxGetApp()->WriteProfileString("BD","Path",m_EditPath);

```
void CPropiedades1Dlg::OnOK()
{
	// TODO: Add extra validation here
	UpdateData(TRUE);

	AfxGetApp()->WriteProfileStrin g ("Leyenda","Salida1",m_EditSalida1);
	AfxGetApp()->WriteProfileString ("Leyenda","Salida2",m_EditSalida2);
	AfxGetApp()->WriteProfileString ("Leyenda","Salida3",m_EditSalida3);
	AfxGetApp()->WriteProfileString ("Leyenda","Salida4",m_EditSalida4);
	AfxGetApp()->WriteProfileString ("Leyenda","Salida5",m_EditSalida5);
	AfxGetApp()->WriteProfileString ("Leyenda","Salida6",m_EditSalida6);
```

```
        AfxGetApp()->WriteProfileString ("Leyenda","Salida7",m_EditSalida7);
        AfxGetApp()->WriteProfileString ("Leyenda","Salida8",m_EditSalida8);

        AfxGetApp()->WriteProfileString ("Leyenda","Entrada1",m_EditEntrada1);
        AfxGetApp()->WriteProfileString ("Leyenda","Entrada2",m_EditEntrada2);
        AfxGetApp()->WriteProfileString ("Leyenda","Entrada3",m_EditEntrada3);
        AfxGetApp()->WriteProfileString ("Leyenda","Entrada4",m_EditEntrada4);

        AfxGetApp()->WriteProfileString ("Socket","IP",m_EditIP);
        AfxGetApp()->WriteProfileString ("Socket","Port",m_EditPort);

        AfxGetApp()->WriteProfileString ("BD","Path",m_EditPath);

        CDialog::OnOK();
}
```

-Insertar función *OnInitDialog();*

(Classwizard -> MessageMap)
(CPropiedades1Dlg -> WM_INITDIALOG ->Add Function->EditCode)

m_EditSalida1=AfxGetApp->GetProfileString("Leyenda","Salida1");
"
m_EditPath=AfxGetApp->GetProfileString("BD","Path");
UpdateData(FALSE);

```
BOOL CPropiedades1Dlg::OnInitDialog()
{
        CDialog::OnInitDialog();

        // TODO: Add extra initialization here

        m_EditSalida1=AfxGetApp()->GetProfileString ("Leyenda","Salida1");
        m_EditSalida2=AfxGetApp()->GetProfileString ("Leyenda","Salida2");
        m_EditSalida3=AfxGetApp()->GetProfileString ("Leyenda","Salida3");
        m_EditSalida4=AfxGetApp()->GetProfileString ("Leyenda","Salida4");
        m_EditSalida5=AfxGetApp()->GetProfileString ("Leyenda","Salida5");
        m_EditSalida6=AfxGetApp()->GetProfileString ("Leyenda","Salida6");
        m_EditSalida7=AfxGetApp()->GetProfileString ("Leyenda","Salida7");
        m_EditSalida8=AfxGetApp()->GetProfileString ("Leyenda","Salida8");

        m_EditEntrada1=AfxGetApp()->GetProfileString ("Leyenda","Entrada1");
        m_EditEntrada2=AfxGetApp()->GetProfileString ("Leyenda","Entrada2");
        m_EditEntrada3=AfxGetApp()->GetProfileString ("Leyenda","Entrada3");
        m_EditEntrada4=AfxGetApp()->GetProfileString ("Leyenda","Entrada4");

        m_EditIP=AfxGetApp()->GetProfileString ("Socket","IP");
        m_EditPort=AfxGetApp()->GetProfileString ("Socket","Port");

        m_EditPath=AfxGetApp()->GetProfileString ("BD","Path");

        UpdateData(FALSE);

        return TRUE;  // return TRUE unless you set the focus to a control
                      // EXCEPTION: OCX Property Pages should return FALSE
}
```

-En *ProyectoBDDlg.h*

-Insertar en la cabecera del fichero

#include "Propiedades1Dlg.h"

```
#if !defined(AFX_PROYECTOBDDLG_H__14710EF7_3ED3_43D7_BACE_258C8EA5D161__INCLUDED_)
#define AFX_PROYECTOBDDLG_H__14710EF7_3ED3_43D7_BACE_258C8EA5D161__INCLUDED_

#include "BaseDatosSet.h"
#include "PuertoParalelo.h"   // Added by ClassView
#include "BaseDatosDlg.h"     // Added by ClassView
```

```
#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

/////////////////////////////////////////////////////////////////////////////
// CProyectoBDDlg dialog
class CBaseDatosSet;
class CBaseDatosDlg;
#include "Propiedades1Dlg.h"
```

-Crear variable en *ProyectoBDDlg.h*

CPropiedades1Dlg Propiedades1Dlg;

```
// Construction
public:
	CProyectoBDDlg(CWnd* pParent = NULL);	// standard constructor

	// incicio de las declaraciones

	void EnviarLeyendaAlSck();
	void ImprimirLeyenda();
	int m_nTimer1;
	void LeerEntradasPuerto();
	void EnviarEstadoAlSck();
	void GuardarSalidas();
	CString Leer(CEdit *Edit);
	CString MirarEstado(int nSal);
	void InicializarSalidas();
	void InsertarRegistro(CString Nombre,CString Estado, CString Hora);
	CString InsertarHora();
	void EscribirRegistro ();

	void ActualizarPuerto();


	// fin de las declaraciones

	CPropiedades1Dlg Propiedades1Dlg;

	CPuertoParalelo PuertoParalelo;
```

-Insertar función *OnAccinPropiedades()*

(Classwizard -> MessageMap)
(CProyectoBDDlg -> ID_ACCIN_PROPIEDADES-> COMMAND
->Add Function->EditCode)

CString BD
BD=AfxGetApp()->GetProfileString("BD","Path");
Propiedades1.Dlg.DoModal();
ImprimirLeyenda();
if (atoi(Leer(&m_EditPort))!=(int)m_Socket1.GetLocalPort())
{
m_Socket1.ABRConectarSck();
m_Socket1.SetLocalPort(atoi(leer(&m_EditPort)));
m_Socket1.ABREscucharSck();
m_Static1=m_Socket1.ABRConvertirEstado(m_Socket1.GetState());
UpdateData(FALSE);
}
if(BD!=AfxGetApp()->GetProfileString("BD","Path")) exit(1);

```
void CProyectoBDDlg::OnAccinPropiedades()
{
	// TODO: Add your command handler code here
	CString BD;
```

```
	BD=AfxGetApp()->GetProfileString ("BD","Path");

	Propiedades1Dlg.DoModal ();
	//While (Propiedades1Dlg.IsWindowVisible ())

	ImprimirLeyenda();

	if (atoi(Leer(&m_EditPort))!= (int) m_Socket1.GetLocalPort ())
	{
		m_Socket1.ABRCerrarSck ();
		m_Socket1.SetLocalPort (atoi(Leer(&m_EditPort)));
		m_Socket1.ABREscucharSck ();
		m_Static1=m_Socket1.ABRConvertirEstado (m_Socket1.GetState ());
		UpdateData(FALSE);
	}
	if (BD!=AfxGetApp()->GetProfileString ("BD","Path")) exit(1);
}
```

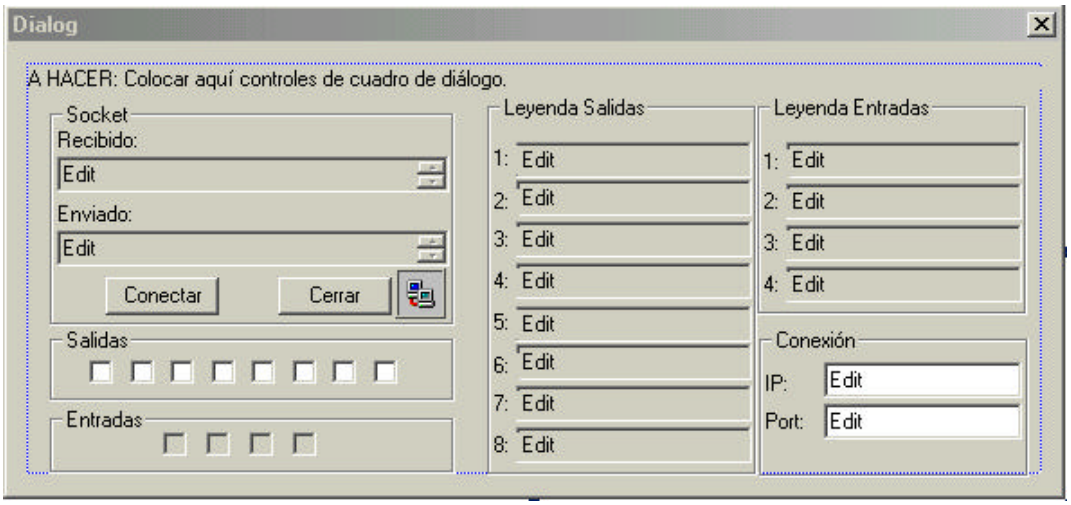
Realizar CTRL+F5 para compilar hasta aquí el proyecto

Añadir a CProyectoBDDlg un diálogo cliente

-Crear dialogo cliente (*IDD_CLIENTEDLG_DIALOG*)
Insert->New Class->Form class->CClienteDlg

-Borrar botones Ok y Cancel

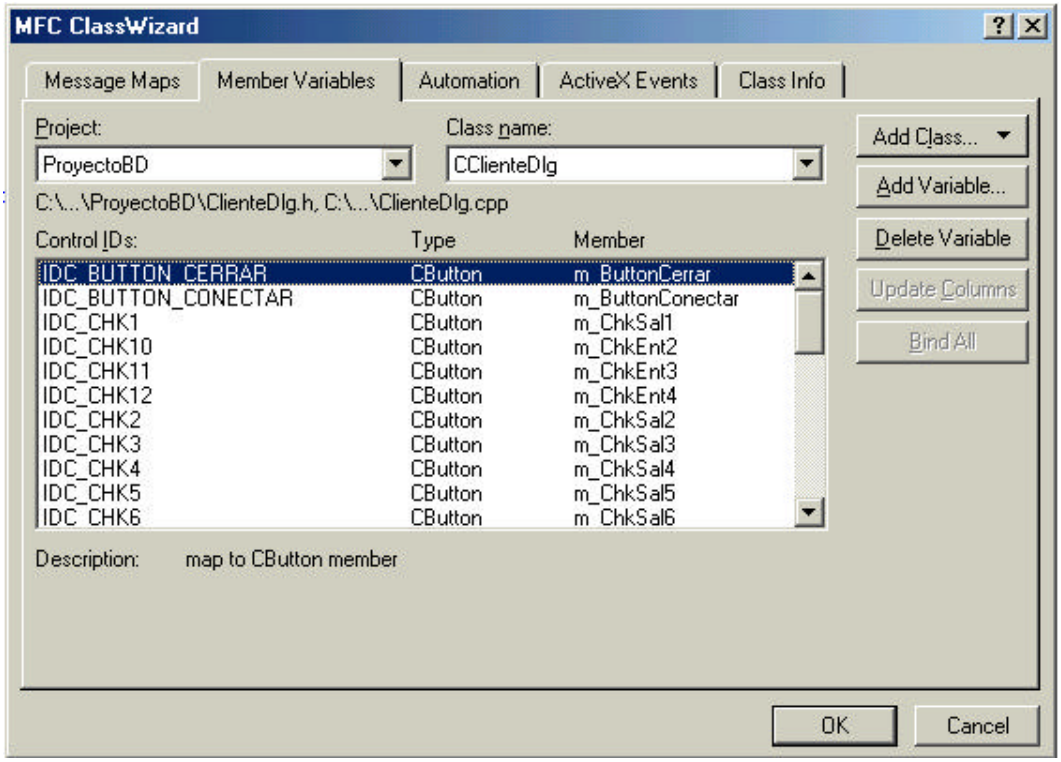
-Insertar los mismos controles que en el dialogo servidor *IDD_ProyectoBD_DIALOG* tal y como se muestra en la siguiente figura



-Crear variables (ClassWizard->Member variables)

IDC_CHK1	*CButton*	*m_ChkSal1*	
	"		
IDC_CHK8	*CButton*	*m_ChkSal8*	
IDC_CHK9	*CButton*	*m_ChkEnt1*	
	"		
IDC_CHK12	*CButton*	*m_ChkEnt4*	
IDC_EDIT01	*CEdit*	*m_EditSalida1*	
	"		
IDC_EDIT08	*CEdit*	*m_EditSalida8*	
IDC_EDIT09	*CEdit*	*m_EditEntrada1*	
	"		
IDC_EDIT012	*CEdit*	*m_EditEntrada4*	
IDC_EDIT013	*CEdit*	*m_EditIP*	
IDC_EDIT014	*CEdit*	*m_EditPort*	
IDC_EDIT016	*CEdit*	*m_EditRecibido*	(Propiedades Read Only, Multiline,VerticalScroll Auto HScroll)
IDC_EDIT017	*CEdit*	*m_EditEnviado*	(Propiedades Read Only, Multiline,VerticalScroll Auto HScroll)

IDC_BUTTON_CONECTAR	*CButton*	*m_ButtonConectar*
IDC_BUTTON_CERRAR	*CButton*	*m_ButtonCerrar*
IDC_STATIC1	*CString*	*m_Static1*
IDC_WINSOCK1	*CMSWinsockControl*	*m_Socket2*



-Crear variables en *ClienteDlg.h*

int Salida1 ... int Salida8
int Entrada1 ... int Entrada4

```
class CClienteDlg : public CDialog
{
// Construction
public:
	CClienteDlg(CWnd* pParent = NULL);   // standard constructor

	int Salida1;
	int Salida2;
	int Salida3;
	int Salida4;
	int Salida5;
	int Salida6;
	int Salida7;
	int Salida8;
	int Entrada1;
	int Entrada2;
	int Entrada3;
	int Entrada4;
```

Preparar la clase CClienteDlg para utilizar funciones propias

-Añadir a *CClienteDlg.h* las declaraciones de las funciones *ABRCClienteDlgUtils.h,* funciones útiles para el diálogo cliente.

```
/////////////////////////////////////////////////////////////////////////////
// CClienteDlg dialog

class CClienteDlg : public CDialog
{
// Construction
public:
	CClienteDlg(CWnd* pParent = NULL);   // standard constructor

	//Inicio de las declaraciones
	void InicializarLeyenda();
	void ActualizarPuerto();
	void InicializarSalidas();
	void Escribir(CEdit* Edit, CString String);
	CString Leer(CEdit* Edit);
	//Fin de las declaraciones

	int Salida1;
```

-Añadir a *ClienteDlg.cpp*

#include "ABRCCLienteDlgUtils.h"

```
// ClienteDlg.cpp : implementation file
//

#include "stdafx.h"
#include "ProyectoBD.h"
#include "ClienteDlg.h"

#include "ABRCClienteDlgUtils.h"

#ifdef _DEBUG
```

Debe estar copiado el fichero *ABRCCLienteDlgUtils.h* en la carpeta del proyecto, fichero donde se encuentran las funciones útiles para el diálogo cliente.

Project->Add to Project->Files->*ABRCClienteDlgUtils.h*

-En *ClienteDlg.cpp*

-Añadir función *OnInitDialog();*

(Classwizard -> MessageMap)

(CClienteDlg -> WM_INITDIALOG ->Add Function->EditCode)

```
m_Static1=m_Socket2.ABRConvertirEstado(m_Socket2.GetState());
UpdateDate(FALSE);
if (AfxGetApp()->GetProfileString("Socket2","IP")=="") AfxGetApp()->Write
ProfileString ("Socket2","IP","127.0.0.1");
if (AfxGetApp()->GetProfileString("Socket2","Port")=="") AfxGetApp()->Write
ProfileString ("Socket2","Port","888");
m_ButtonCerrar.EnableWindow(FALSE);
m_ButtonConectar.EnableWindow(TRUE);
Salida1=0;
```

"

Salida8=0;
return TRUE

```
BOOL CClienteDlg::OnInitDialog()
{
	CDialog::OnInitDialog();

	// TODO: Add extra initialization here
	m_Static1=m_Socket2.ABRConvertirEstado (m_Socket2.GetState ());
	UpdateData(FALSE);

	if (AfxGetApp()->GetProfileString ("Socket2","IP")=="")    AfxGetApp()-
>WriteProfileString ("Socket2","IP","127.0.0.1");
	if (AfxGetApp()->GetProfileString ("Socket2","Port")=="") AfxGetApp()-
>WriteProfileString ("Socket2","Port","888");


	m_EditIP.SetWindowText (AfxGetApp()->GetProfileString ("Socket2","IP"));
	m_EditPort.SetWindowText (AfxGetApp()->GetProfileString ("Socket2","Port"));

	m_ButtonCerrar.EnableWindow (FALSE);
	m_ButtonConectar.EnableWindow (TRUE);

	Salida1=0;
	Salida2=0;
	Salida3=0;
	Salida4=0;
	Salida5=0;
	Salida6=0;
	Salida7=0;
	Salida8=0;

	return TRUE;  // return TRUE unless you set the focus to a control
		      // EXCEPTION: OCX Property Pages should return FALSE
}
```

-Añadir función *OnButtonConectar()*
(Classwizard -> MessageMap)
(CClienteDlg -> IDC_BUTTON_CONECTAR->BN_CLICKED
Add Function-> ->EditCode)

m_Socket2.ABRCerrarSck();
m_Socket2.ABR_ConectarSck();
m_Static1=m_Socket2.ABRConvertirEstado...
UpdateData(FALSE);
m_ButtonCerrar.EnableWindow(TRUE);
m_ButtonConectar.EnableWindow(FALSE);

```
void CClienteDlg::OnButtonConectar()
{
	// TODO: Add your control notification handler code here
	m_Socket2.ABRCerrarSck ();
	m_Socket2.ABRConectarSck ();
	m_Static1=m_Socket2.ABRConvertirEstado (m_Socket2.GetState ());
	UpdateData(FALSE);
	m_ButtonCerrar.EnableWindow (TRUE);
	m_ButtonConectar.EnableWindow (FALSE);
}
```

-Añadir funcion *OnChangeEdit013()*
(Classwizard -> MessageMap)
(CClienteDlg -> IDC_EDIT013->EN_CHANGE
Add Function-> ->EditCode)

AfxGetApp()->WriteProfileString("Socket2","IP",Leer(&m_EditIP))

```
void CClienteDlg::OnChang eEdit013()
{
	// TODO: If this is a RICHEDIT control, the control will not
	// send this notification unless you override the CDialog::OnInitDialog()
	// function and call CRichEditCtrl().SetEventMask()
	// with the ENM_CHANGE flag ORed into the mask.

	// TODO: Add your control notification handler code here
	AfxGetApp()->WriteProfileString ("Socket2","IP",Leer(&m_EditIP));
}
```

-Añadir funcion *OnChangeEdit014()*
(Classwizard -> MessageMap)
(CClienteDlg -> IDC_EDIT014->EN_CHANGE
Add Function-> ->EditCode)

AfxGetApp()->WriteProfileString("Socket2","Port",Leer(&m_EditPort))

```
void CClienteDlg::OnChangeEdit014()
{
	// TODO: If this is a RICHEDIT control, the control will not
	// send this notification unless you override the CDialog::OnInitDialog()
	// function and call CRichEditCtrl().SetEventMask()
	// with the ENM_CHANGE flag ORed into the mask.

	// TODO: Add your control notification handler code here
	AfxGetApp()->WriteProfileString ("Socket2","Port",Leer(&m_EditPort));

}
```

-Añadir funcion *OnConnectWinsock1()*
(Classwizard -> MessageMap)
(CClienteDlg -> IDC_WINSOCK1->Connect
Add Function-> ->EditCode)

m_Static1=m_Socket2.ABRConvertirEstado ...
UpdateData(FALSE);

```
void CClienteDlg::OnConnectWinsock1()
{
	// TODO: Add your control notification handler code here
	m_Static1=m_Socket2.ABRConvertirEstado (m_Socket2.GetState ());
	UpdateData(FALSE);

}
```

-Añadir función *OnButtonCerrar()*
(Classwizard -> MessageMap)
(CClienteDlg -> IDC_BUTTON_CERRAR->BN_CLICKED
Add Function-> ->EditCode)

mSocket2.ABRCerrarSck();
m_Static1=m_Socket2.ABRConvertirEstado(m_Socket2.GetState());
UpdateData(FALSE);
m_ButtonCerrar.EnableWindow(FALSE);
m_ButtonConectar.EnableWindow(TRUE);

```
void CClienteDlg::OnButtonCerrar()
{
```

```
        // TODO: Add your control notification handler code here
        m_Socket2.ABRCerrarSck ();
        m_Static1=m_Socket2.ABRConvertirEstado (m_Socket2.GetState ());
        UpdateData(FALSE);
        m_ButtonCerrar.EnableWindow (FALSE);
        m_ButtonConectar.EnableWindow (TRUE);
}
```

-Añadir función *OnDataArrivalWinsock1()*

(Classwizard -> MessageMap)
(CClienteDlg -> IDC_WINSOCK1->DataArrival
Add Function-> ->EditCode)

m_Socket2.ABROnDataArrivalWinsock1(bytestotal);
CString aux;
aux = m_Socket2.Dato;
aux = aux.Mid(0,7) ;
if (aux=="Estado:") InicializarSalidas();
if (aux=="Leyenda") Inicializar Leyenda ();
else Escribir (&m_EditRecibido,m_Socket2.Dato);

```
void CClienteDlg::OnDataArrivalWinsock1(long bytesTotal)
{
        // TODO: Add your control notification handler code here
        m_Socket2.ABROnDataArrivalWinsock1 (bytesTotal);


        CString aux;
        aux=m_Socket2.Dato;
        aux=aux.Mid (0,7);

        if (aux=="Estado:") InicializarSalidas();
        if (aux=="Leyenda") InicializarLeyenda();
        else Escribir(&m_EditRecibido,m_Socket2.Dato );

}
```

-Añadir funciones *OnChk1() ... OnChk8()*

OnChk1()
(Classwizard -> MessageMap)
(CClienteDlg -> IDC_CHK1 ->BN_CLICKED
Add Function-> ->EditCode)

OnChk2()
(Classwizard -> MessageMap)
(CClienteDlg -> IDC_CHK2 ->BN_CLICKED
Add Function-> ->EditCode)

OnChk3()
(Classwizard -> MessageMap)
(CClienteDlg -> IDC_CHK3 ->BN_CLICKED
Add Function-> ->EditCode)

OnChk4()
(Classwizard -> MessageMap)
(CClienteDlg -> IDC_CHK4 ->BN_CLICKED
Add Function-> ->EditCode)

OnChk5()
(Classwizard -> MessageMap)
(CClienteDlg -> IDC_CHK5 ->BN_CLICKED
Add Function-> ->EditCode)

OnChk6()
(Classwizard -> MessageMap)
(CClienteDlg -> IDC_CHK6 ->BN_CLICKED
Add Function-> ->EditCode)

OnChk7()
(Classwizard -> MessageMap)
(CClienteDlg -> IDC_CHK7 ->BN_CLICKED
Add Function-> ->EditCode)

OnChk8()
(Classwizard -> MessageMap)
(CClienteDlg -> IDC_CHK8 ->BN_CLICKED
Add Function-> ->EditCode)

Copiar el código de las funciones anteriores *ClienteDlg.cpp*

```
void CClienteDlg::OnChk1()
{
	// TODO: Add your control notification handler code here
	if (m_Socket2.GetState ()==7)
	{
		Salida1=!Salida1;
		m_Socket2.ABREnviarSck ("Salida1");
		Escribir(&m_EditEnviado,Leer(&m_EditSalida1));
	}
	else m_ChkSal1.SetCheck (0);
}

void CClienteDlg::OnChk2()
{
	// TODO: Add your control notification handler code here
	if (m_Socket2.GetState ()==7)
	{
		Salida2=!Salida2;
		m_Socket2.ABREnviarSck ("Salida2");
		Escribir(&m_EditEnviado,Leer(&m_EditSalida2));
	}
	else m_ChkSal2.SetCheck (0);
}

void CClienteDlg::OnChk3()
{
	// TODO: Add your control notification handler code here
	if (m_Socket2.GetState ()==7)
	{
		Salida3=!Salida3;
		m_Socket2.ABREnviarSck ("Salida3");
		Escribir(&m_EditEnviado,Leer(&m_EditSalida3));
	}
	else m_ChkSal3.SetCheck (0);
}

void CClienteDlg::OnChk4()
{
	// TODO: Add your control notification handler code here
	if (m_Socket2.GetState ()==7)
```

```
	{
		Salida4=!Salida4;
		m_Socket2.ABREnviarSck ("Salida4");
		Escribir(&m_EditEnviado,Leer(&m_EditSalida4));
	}
	else m_ChkSal4.SetCheck (0);
}

void CClienteDlg::OnChk5()
{
	// TODO: Add your control notification handler code here
	if (m_Socket2.GetState ()==7)
	{
		Salida5=!Salida5;
		m_Socket2.ABREnviarSck ("Salida5");
		Escribir(&m_EditEnviado,Leer(&m_EditSalida5));
	}
	else m_ChkSal5.SetCheck (0);
}

void CClienteDlg::OnChk6()
{
	// TODO: Add your control notification handler code here
	if (m_Socket2.GetState ()==7)
	{
		Salida6=!Salida6;
		m_Socket2.ABREnviarSck ("Salida6");
		Escribir(&m_EditEnviado,Leer(&m_EditSalida6));
	}
	else m_ChkSal6.SetCheck (0);
}

void CClienteDlg::OnChk7()
{
	// TODO: Add your control notification handler code here
	if (m_Socket2.GetState ()==7)
	{
		Salida7=!Salida7;
		m_Socket2.ABREnviarSck ("Salida7");
		Escribir(&m_EditEnviado,Leer(&m_EditSalida7));
	}
	else m_ChkSal7.SetCheck (0);
}

void CClienteDlg::OnChk8()
{
	// TODO: Add your control notification handler code here
	if (m_Socket2.GetState ()==7)
	{
		Salida8=!Salida8;
		m_Socket2.ABREnviarSck ("Salida8");
		Escribir(&m_EditEnviado,Leer(&m_EditSalida8));
	}
	else m_ChkSal8.SetCheck (0);
}
```

-Añadir función *OnCloseWinsock1()*

(Classwizard -> MessageMap)
(CClienteDlg -> IDC_WINSOCK1->Close -> Add Function-> EditCode)

m_Socket2.ABRCerrarSck();
m_Static1=ABRConvertirEstado...
UpdateData(FALSE);
m_ButtonCerrar.EnableWindow(FALSE);
m_ButtonConectar.EnableWindow(TRUE);

```
void CClienteDlg::OnCloseWinsock1()
{
	// TODO: Add your control notification handler code here
	m_Socket2.ABRCerrarSck ();
	m_Static1=m_Socket2.ABRConvertirEstado (m_Socket2.GetState ());
	UpdateData(FALSE);
```

```
        m_ButtonCerrar.EnableWindow (FALSE);
        m_ButtonConectar.EnableWindow (TRUE);
}
```

-En *ProyectoBDDlg.h*

-Insertar en la cabecera del fichero
#include "ClienteDlg.h"

```
#if !defined(AFX_PROYECTOBDDLG_H__14710EF7_3ED3_43D7_BACE_258C8EA5D161__INCLUDED_)
#define AFX_PROYECTOBDDLG_H__14710EF7_3ED3_43D7_BACE_258C8EA5D161__INCLUDED_

#include "BaseDatosSet.h"
#include "PuertoParalelo.h"   // Added by ClassView
#include "BaseDatosDlg.h"     // Added by ClassView

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

/////////////////////////////////////////////////////////////////////////////
// CProyectoBDDlg dialog
class CBaseDatosSet;
class CBaseDatosDlg;
#include "Propiedades1Dlg.h"
#include "ClienteDlg.h"
```

-Añadir variable en *ProyectoBDDlg.h*
CClienteDlg ClienteDlg;

```
// Construction
public:
        CProyectoBDDlg(CWnd* pParent = NULL);       // standard constructor

        // incicio de las declaraciones

        void EnviarLeyendaAlSck();
        void ImprimirLeyenda();
        int m_nTimer1;
        void LeerEntradasPuerto();
        void EnviarEstadoAlSck();
        void GuardarSalidas();
        CString Leer(CEdit *Edit);
        CString MirarEstado(int nSal);
        void InicializarSalidas();
        void InsertarRegistro(CString Nombre,CString Estado, CString Hora);
        CString InsertarHora();
        void EscribirRegistro ();

        void ActualizarPuerto();


        // fin de las declaraciones

        CClienteDlg ClienteDlg;

        CPropiedades1Dlg Propiedades1Dlg;

        CPuertoParalelo PuertoParalelo;

        int Salida1;
```

-Añadir la función *OnAccinCliente()* en *CProyectoBDDlg*
(Classwizard -> MessageMap)
(CProyectoBDDlg -> ID_ACCIN_CLIENTE->COMMAND
Add Function-> ->EditCode)

ClienteDlg.DoModal();

```
void CProyectoBDDlg::OnAccinCliente()
{
	// TODO: Add your command handler code here
	ClienteDlg.DoModal ();
}
```

Realizar CTRL+F5 para compilar hasta aquí el proyecto

Ficheros que incluimos en el proyecto

A continuación vamos a mostrar los ficheros que deberemos incluir en el proyecto para utilizar funciones propias útiles para las diferentes partes del proyecto.

PuertoParalelo.cpp	
PuertoParalelo.h	Clase que controla el puerto paralelo
ABRdb.h	Funciones para gestionar la base de datos
ABRCBaseDatosDlgUtils.h	Funciones útiles para el diálogo base de datos
ABRCProyectoBDDlgUtils.h	Funciones útiles para el diálogo principal
ABRWinsock.h	Funciones útiles gestionar el control Winsock
ABRCClienteDlgUtils.h	Funciones útiles para el diálogo cliente

PuertoParalelo.cpp y PuertoParalelo.h, clase que controla el puerto paralelo

"PuertoParalelo.cpp"

```
// PuertoParalelo.cpp: implementation of the CPuertoParalelo class.
//
//////////////////////////////////////////////////////////////////////

#include "stdafx.h"
#include "PuertoParalelo.h"

#ifdef _DEBUG
#undef THIS_FILE
static char THIS_FILE[]=__FILE__;
#define new DEBUG_NEW
#endif

//////////////////////////////////////////////////////////////////////
// Construction/Destruction
//////////////////////////////////////////////////////////////////////

CPuertoParalelo::CPuertoParalelo()
{

}

CPuertoParalelo::~CPuertoParalelo()
{

}


int CPuertoParalelo::Leer()
{
	return _inp(0x379);
}

void CPuertoParalelo::Escribir(int dato)
{
	_outp(0x378,dato);
}

void CPuertoParalelo::Salida1(bool estado)
{
	if (estado) _outp(0x378,_inp(0x378)|0x01);
	else _outp(0x378,_inp(0x378)&(~0x01));
}

void CPuertoParalelo::Salida2(bool estado)
{
	if (estado) _outp(0x378,_inp(0x378)|0x02);
	else _outp(0x378,_inp(0x378)&(~0x02));
}

void CPuertoParalelo::Salida3(bool estado)
{
	if (estado) _outp(0x378,_inp(0x378)|0x04);
	else _outp(0x378,_inp(0x378)&(~0x04));
}

void CPuertoParalelo::Salida4(bool estado)
{
	if (estado) _outp(0x378,_inp(0x378)|0x08);
	else _outp(0x378,_inp(0x378)&(~0x08));
}

void CPuertoParalelo::Salida5(bool estado)
{
	if (estado) _outp(0x378,_inp(0x378)|0x10);
	else _outp(0x378,_inp(0x378)&(~0x10));
}

void CPuertoParalelo::Salida6(bool estado)
```

```
{
	if (estado) _outp(0x378,_inp(0x378)|0x20);
	else _outp(0x378,_inp(0x378)&(~0x20));
}

void CPuertoParalelo::Salida7(bool estado)
{
	if (estado) _outp(0x378,_inp(0x378)|0x40);
	else _outp(0x378,_inp(0x378)&(~0x40));
}

void CPuertoParalelo::Salida8(bool estado)
{
	if (estado) _outp(0x378,_inp(0x378)|0x80);
	else _outp(0x378,_inp(0x378)&(~0x80));
}

int CPuertoParalelo::Entrada1()
{
	return ((_inp(0x379)&0x08)>>3);
}
int CPuertoParalelo::Entrada2()
{
	return ((_inp(0x379)&0x10)>>4);
}
int CPuertoParalelo::Entrada3()
{
	return ((_inp(0x379)&0x20)>>5);
}
int CPuertoParalelo::Entrada4()
{
	return ((_inp(0x379)&0x40)>>6);
}
```

"PuertoParalelo.h"

```
// PuertoParalelo.h: interface for the CPuertoParalelo class.
//
//////////////////////////////////////////////////////////////////////

#if !defined(AFX_PUERTOPARALELO_H__4E5D8741_A1E6_11D5_97E1_A19932F01649__INCLUDED_)
#define AFX_PUERTOPARALELO_H__4E5D8741_A1E6_11D5_97E1_A19932F01649__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

#include <conio.h>     //Libreria donde se encuentran las funciones
                                    // _inp() y _outp()

class CPuertoParalelo
{
public:
	CPuertoParalelo();
	virtual ~CPuertoParalelo();

	int Entrada1();
	int Entrada2();
	int Entrada3();
	int Entrada4();
	void Salida1(bool estado);
	void Salida2(bool estado);
	void Salida3(bool estado);
	void Salida4(bool estado);
	void Salida5(bool estado);
	void Salida6(bool estado);
	void Salida7(bool estado);
	void Salida8(bool estado);
	void Escribir(int dato);
	int Leer();
};

#endif //
!defined(AFX_PUERTOPARALELO_H__4E5D8741_A1E6_11D5_97E1_A19932F01649__INCLUDED_)
```

ABRdb.h, funciones para gestionar la base de datos

```
void CBaseDatosSet::ABRAbrirBD()
{
//      m_pSet=&m_BaseDatosDlgSet;
        if (IsOpen ()){
                Close ();
        }
        Open ();

}

void CBaseDatosSet::ABRCerrarBD()
{
        if (IsOpen ()){
                Close ();
        }
}

void CBaseDatosSet::ABRUltimoRegistro()
{
        if ((IsOpen ())&&(!IsEOF())){
                MoveLast();
        }

}

void CBaseDatosSet::ABRPrimerRegistro()
{
        if ((IsOpen ())&&(!IsBOF())){
                MoveFirst();
        }

}

void CBaseDatosSet::ABRSiguienteRegistro()
{

        if ((IsOpen ())&&(!IsEOF())&&(IsBOF())){
                MoveNext();
        }
        if ((IsOpen ())&&(!IsEOF())){
                MoveNext();
        }

}

void CBaseDatosSet::ABRAnteriorRegistro()
{
        if ((IsOpen ())&&(!IsBOF())&&(IsEOF())){
                MovePrev();
        }
        if ((IsOpen ())&&(!IsBOF())){
                MovePrev();
        }
}

void CBaseDatosSet::ABRInsertarRegistro(CString Nombre,CString Estado,CString Hora)
{
        CString sqlcmd;

//      UpdateData(TRUE);

        if (IsOpen ()){
                sqlcmd.Format("INSERT INTO Historico (Nombre,Estado,Hora) VALUES
('%s','%s','%s')",Nombre,Estado,Hora);
                m_pDatabase->ExecuteSQL (sqlcmd);
//              ABRBuscarTodosLosRegistros();
        }

}

void CBaseDatosSet::ABRBorrarRegistro(CString Nombre,CString Estado,CString Hora)
{
```

```
	CString sqlcmd;

//	UpdateData(TRUE);

	CString n="=";
	CString e="=";
	CString h="=";

	if (Nombre=="") n="<>";
	if (Estado=="") e="<>";
	if (Hora=="") h="<>";



	if (IsOpen ()){
		sqlcmd.Format("DELETE FROM Historico WHERE (Nombre %s '%s' AND Estado %s
'%s' AND Hora %s '%s')",n,Nombre,e,Estado,h,Hora);

		if ((n==e) && (e==h) && (h=="<>"))
		{
			if (AfxMessageBox("¿Seguro que desea Borrar Todos Los Registros?",
MB_YESNO)==IDYES)
			{
				m_pDatabase->ExecuteSQL (sqlcmd);
			}

		}
		else m_pDatabase->ExecuteSQL (sqlcmd);

//		LimpiarCampos();

//		ABRBuscarTodosLosRegistros();
	}

}

void CBaseDatosSet::ABRActualizarRegistro(CString Nombre,CString Estado,CString Hora)
{
	CString sqlcmd;

//	UpdateData(TRUE);
	if (IsOpen ()){
		sqlcmd.Format("UPDATE Historico SET Nombre='%s',Estado='%s',Hora='%s'
WHERE (Nombre='%s'AND Estado='%s'AND Hora='%s')",Nombre, Estado, Hora, m_Nombre,
m_Estado, m_Hora);
		m_pDatabase->ExecuteSQL (sqlcmd);
//		ABRBuscarTodosLosRegistros();
	}

}

void CBaseDatosSet::ABRBuscarRegistro(CString Nombre,CString Estado,CString Hora)
{
	CString sqlcmd,aux;

//	UpdateData(TRUE);

	CString n="=";
	CString e="=";
	CString h="=";

	if (Nombre=="") n="<>";
	if (Estado=="") e="<>";
	if (Hora=="") h="<>";

	if (IsOpen ()){

//		sqlcmd.Format("SELECT COUNT (Hora) FROM Historico WHERE (Nombre %s '%s'
AND Estado %s '%s' AND Hora %s '%s')",n,Nombre,e,Estado,h,Hora);
//		ABRCerrarBD();
//		Open (0,sqlcmd,NULL);

//		aux.Format ("%s Registros encontrados.",m_pSet->m_Nombre);
//		m_Static1=aux;
//		UpdateData(FALSE);
```

```
                sqlcmd.Format("SELECT * FROM Historico WHERE (Nombre %s '%s' AND Estado
%s '%s' AND Hora %s '%s')",n,Nombre,e,Estado,h,Hora);
                //LimpiarCampos();
                ABRCerrarBD();
                Open (0,sqlcmd,NULL);

//              UltimoRegistro();
//              LeerRegistro();
        }

}

void CBaseDatosSet::ABRBuscarTodosLosRegistros()
{
        CString sqlcmd;

        if (IsOpen ()){

                sqlcmd.Format("SELECT * FROM Historico");
                ABRCerrarBD();
                Open (0,sqlcmd,NULL);

                ABRUltimoRegistro();
        }

}


CString CBaseDatosSet::ABRContarRegistros()
{
        CString sqlcmd;
        CString aux;

        if (IsOpen ()){

                sqlcmd.Format("SELECT COUNT (Hora) FROM Historico");
                ABRCerrarBD();
                Open (0,sqlcmd,NULL);
                aux.Format ("%s Registros en la Base de Datos.",m_Nombre);

                sqlcmd.Format("SELECT * FROM Historico");
                ABRCerrarBD();
                Open (0,sqlcmd,NULL);

        }

        return aux;
}
```

ABRCBaseDatosDlgUtils.h, funciones útiles para el diálogo base de datos

```
CString CBaseDatosDlg::InsertarHora()
{
	CString Hora;
	COleDateTime date=COleDateTime::GetCurrentTime ();

	Hora.Format ("%.2d/%.2d/%d
%.2d:%.2d:%.2d",date.GetDay(),date.GetMonth(),date.GetYear (),date.GetHour
(),date.GetMinute (),date.GetSecond ());

	return Hora;
}

void CBaseDatosDlg::Escribir(CEdit *Edit, CString String)
{
	CString aux;

	Edit->GetWindowText (aux);
	aux.Insert (aux.GetLength (),String);
	aux.Insert (aux.GetLength (),"\r\n");
	Edit->SetWindowText (aux);
	Edit->SetSel (Edit->GetWindowTextLength (),Edit->GetWindowTextLength (),TRUE);
	Edit->LineScroll (Edit->GetScrollLimit(SB_VERT));

	UpdateData(FALSE);
}

CString CBaseDatosDlg::Leer(CEdit *Edit)
{
	CString aux;
	Edit->GetWindowText (aux);
	return aux;
}

void CBaseDatosDlg::EscribirRegistro ()
{
	m_EditNombre.SetWindowText (m_pSet->m_Nombre);
	m_EditEstado.SetWindowText (m_pSet->m_Estado);
	m_EditHora.SetWindowText (m_pSet->m_Hora);
}

void CBaseDatosDlg::BorrarCampos()
{
	m_EditNombre.SetWindowText ("");
	m_EditEstado.SetWindowText ("");
	m_EditHora.SetWindowText ("");
}
```

ABRCProyectoBDDlgUtils.h, funciones útiles para el diálogo principal

```
void CProyectoBDDlg::InsertarRegistro(CString Nombre, CString Estado, CString Hora)
{
CString aux;

	m_pSet->m_Nombre=Nombre;
	m_pSet->m_Estado=Estado;
	m_pSet->m_Hora=Hora;
	EscribirRegistro();


	m_pSet=&m_BaseDatosDlgSet;
	m_pSet->ABRAbrirBD ();

	m_pSet->ABRInsertarRegistro(Nombre,Estado,Hora);

	m_pSet->ABRCerrarBD ();

}

CString CProyectoBDDlg::InsertarHora()
{
	CString Hora;
	COleDateTime date=COleDateTime::GetCurrentTime ();

	Hora.Format ("%.2d/%.2d/%d
%.2d:%.2d:%.2d",date.GetDay(),date.GetMonth(),date.GetYear (),date.GetHour
(),date.GetMinute (),date.GetSecond ());

	return Hora;
}

void CProyectoBDDlg::EscribirRegistro ()
{
	m_EditNombre.SetWindowText (m_pSet->m_Nombre);
	m_EditEstado.SetWindowText (m_pSet->m_Estado);
	m_EditHora.SetWindowText (m_pSet->m_Hora);
}

CString CProyectoBDDlg::Leer(CEdit *Edit)
{
	CString aux;
	Edit->GetWindowText (aux);
	return aux;
}

void CProyectoBDDlg::InicializarSalidas()
{
	CString Estado;

	/*Entrada1=PuertoParalelo.Entrada1 ();
	Entrada2=PuertoParalelo.Entrada2 ();
	Entrada3=PuertoParalelo.Entrada3 ();
	Entrada4=PuertoParalelo.Entrada4 ();
*/
	m_pSet->ABRBuscarRegistro ("Se Cierra Dialogo de Salidas","","");
	if (m_pSet->m_Nombre!="Se Cierra Dialogo de Salidas")
	{
		m_pSet->ABRInsertarRegistro ("Inicio de la Base de Datos","-
",InsertarHora());
		Estado="0 0 0 0 0 0 0 0*0 0 0 0" ;
	}
	else
	{
		m_pSet->ABRUltimoRegistro();
		Estado=m_pSet->m_Estado ;
	}


	if (Estado.Mid (0,1)=="1") Salida1=1;
	else Salida1=0;
	if (Estado.Mid (2,1)=="1") Salida2=1;
	else Salida2=0;
```

```
	if (Estado.Mid (4,1)=="1") Salida3=1;
	else Salida3=0;
	if (Estado.Mid (6,1)=="1") Salida4=1;
	else Salida4=0;
	if (Estado.Mid (8,1)=="1") Salida5=1;
	else Salida5=0;
	if (Estado.Mid (10,1)=="1") Salida6=1;
	else Salida6=0;
	if (Estado.Mid (12,1)=="1") Salida7=1;
	else Salida7=0;
	if (Estado.Mid (14,1)=="1") Salida8=1;
	else Salida8=0;
	if (Estado.Mid (16,1)=="1") Entrada1=1;
	else Entrada1=0;
	if (Estado.Mid (18,1)=="1") Entrada2=1;
	else Entrada2=0;
	if (Estado.Mid (20,1)=="1") Entrada3=1;
	else Entrada3=0;
	if (Estado.Mid (22,1)=="1") Entrada4=1;
	else Entrada4=0;


	m_EditNombre.SetWindowText ("Se abre Dialogo de Salidas");
	m_EditEstado.SetWindowText (Estado);
	m_EditHora.SetWindowText (InsertarHora());

	InsertarRegistro(Leer(&m_EditNombre),Leer(&m_EditEstado),Leer(&m_EditHora));
	ActualizarPuerto();
}

void CProyectoBDDlg::GuardarSalidas()
{
	CString Estado;

	if (Salida1==1)	Estado.Insert (1,"1 ");
	else Estado.Insert (1,"0 ");
	if (Salida2==1)	Estado.Insert (3,"1 ");
	else Estado.Insert (3,"0 ");
	if (Salida3==1)	Estado.Insert (5,"1 ");
	else Estado.Insert (5,"0 ");
	if (Salida4==1)	Estado.Insert (7,"1 ");
	else Estado.Insert (7,"0 ");
	if (Salida5==1)	Estado.Insert (9,"1 ");
	else Estado.Insert (9,"0 ");
	if (Salida6==1)	Estado.Insert (11,"1 ");
	else Estado.Insert (11,"0 ");
	if (Salida7==1)	Estado.Insert (13,"1 ");
	else Estado.Insert (13,"0 ");
	if (Salida8==1)	Estado.Insert (15,"1*");
	else Estado.Insert (15,"0*");
	if (Entrada1==1) Estado.Insert (17,"1 ");
	else Estado.Insert (17,"0 ");
	if (Entrada2==1) Estado.Insert (19,"1 ");
	else Estado.Insert (19,"0 ");
	if (Entrada3==1) Estado.Insert (21,"1 ");
	else Estado.Insert (21,"0 ");
	if (Entrada4==1) Estado.Insert (23,"1");
	else Estado.Insert (23,"0");

	m_EditNombre.SetWindowText ("Se Cierra Dialogo de Salidas");
	m_EditEstado.SetWindowText (Estado);
	m_EditHora.SetWindowText (InsertarHora());

	m_pSet-
>ABRInsertarRegistro(Leer(&m_EditNombre),Leer(&m_EditEstado),Leer(&m_EditHora));

}


CString CProyectoBDDlg::MirarEstado(int nSal)
{
	CString out;

	switch (nSal)
	{
		case 1:
			if (Salida1) out="0";
```

```
                        else out="1";
                        Salida1=!Salida1;
                        break;
                case 2:
                        if (Salida2) out="0";
                        else out="1";
                        Salida2=!Salida2;
                        break;
                case 3:
                        if (Salida3) out="0";
                        else out="1";
                        Salida3=!Salida3;
                        break;
                case 4:
                        if (Salida4) out="0";
                        else out="1";
                        Salida4=!Salida4;
                        break;
                case 5:
                        if (Salida5) out="0";
                        else out="1";
                        Salida5=!Salida5;
                        break;
                case 6:
                        if (Salida6) out="0";
                        else out="1";
                        Salida6=!Salida6;
                        break;
                case 7:
                        if (Salida7) out="0";
                        else out="1";
                        Salida7=!Salida7;
                        break;
                case 8:
                        if (Salida8) out="0";
                        else out="1";
                        Salida8=!Salida8;
                        break;
                case 9:
                        if (!Entrada1) out="0";
                        else out="1";
                        break;
                case 10:
                        if (!Entrada2) out="0";
                        else out="1";
                        break;
                case 11:
                        if (!Entrada3) out="0";
                        else out="1";
                        break;
                case 12:
                        if (!Entrada4) out="0";
                        else out="1";
                        break;
                default:
                        return "NULL";

        }
        return out;
}


void CProyectoBDDlg::ActualizarPuerto()
{
        if (Salida1==1)
        {
                PuertoParalelo.Salida1 (1);
                m_CheckSal1.SetCheck (1);
        }
        else
        {
                PuertoParalelo.Salida1 (0);
                m_CheckSal1.SetCheck (0);
        }
        if (Salida2==1)
        {
```

```
		PuertoParalelo.Salida2 (1);
		m_CheckSal2.SetCheck (1);
	}
	else
	{
		PuertoParalelo.Salida2 (0);
		m_CheckSal2.SetCheck (0);
	}
	if (Salida3==1)
	{
		PuertoParalelo.Salida3 (1);
		m_CheckSal3.SetCheck (1);
	}
	else
	{
		PuertoParalelo.Salida3 (0);
		m_CheckSal3.SetCheck (0);
	}
	if (Salida4==1)
	{
		PuertoParalelo.Salida4 (1);
		m_CheckSal4.SetCheck (1);
	}
	else
	{
		PuertoParalelo.Salida4 (0);
		m_CheckSal4.SetCheck (0);
	}
	if (Salida5==1)
	{
		PuertoParalelo.Salida5 (1);
		m_CheckSal5.SetCheck (1);
	}
	else
	{
		PuertoParalelo.Salida5 (0);
		m_CheckSal5.SetCheck (0);
	}
	if (Salida6==1)
	{
		PuertoParalelo.Salida6 (1);
		m_CheckSal6.SetCheck (1);
	}
	else
	{
		PuertoParalelo.Salida6 (0);
		m_CheckSal6.SetCheck (0);
	}
	if (Salida7==1)
	{
		PuertoParalelo.Salida7 (1);
		m_CheckSal7.SetCheck (1);
	}
	else
	{
		PuertoParalelo.Salida7 (0);
		m_CheckSal7.SetCheck (0);
	}
	if (Salida8==1)
	{
		PuertoParalelo.Salida8 (1);
		m_CheckSal8.SetCheck (1);
	}
	else
	{
		PuertoParalelo.Salida8 (0);
		m_CheckSal8.SetCheck (0);
	}

	// Enviar Nuevo Estado al Socket

	if (m_Socket1.GetState ()==7) EnviarEstadoAlSck();

}

void CProyectoBDDlg::LeerEntradasPuerto()
{
```

```
	int Ent1Aux;
	int Ent2Aux;
	int Ent3Aux;
	int Ent4Aux;
	//Guardar Entradas anteriores
	Ent1Aux=Entrada1;
	Ent2Aux=Entrada2;
	Ent3Aux=Entrada3;
	Ent4Aux=Entrada4;
	//Actualizar las entradas
	Entrada1=PuertoParalelo.Entrada1 ();
	Entrada2=PuertoParalelo.Entrada2 ();
	Entrada3=PuertoParalelo.Entrada3 ();
	Entrada4=PuertoParalelo.Entrada4 ();
	//Actualizar Entradas en el dialogo
	if (Entrada1==1) m_CheckEnt1.SetCheck (1);
	else m_CheckEnt1.SetCheck (0);
	if (Entrada2==1) m_CheckEnt2.SetCheck (1);
	else m_CheckEnt2.SetCheck (0);
	if (Entrada3==1) m_CheckEnt3.SetCheck (1);
	else m_CheckEnt3.SetCheck (0);
	if (Entrada4==1) m_CheckEnt4.SetCheck (1);
	else m_CheckEnt4.SetCheck (0);
	//Registro en la base de datos de los cambios
	if (Ent1Aux!=Entrada1)
	{
		InsertarRegistro(Leer(&m_EditEntrada1),MirarEstado(9),InsertarHora());

	}
	if (Ent2Aux!=Entrada2)
	{
		InsertarRegistro(Leer(&m_EditEntrada2),MirarEstado(10),InsertarHora());

	}
	if (Ent3Aux!=Entrada3)
	{
		InsertarRegistro(Leer(&m_EditEntrada3),MirarEstado(11),InsertarHora());

	}
	if (Ent4Aux!=Entrada4)
	{
		InsertarRegistro(Leer(&m_EditEntrada4),MirarEstado(12),InsertarHora());

	}
	// Enviar Estado al socket si hay cambios
	if((Ent1Aux!=Entrada1)||(Ent2Aux!=Entrada2)||(Ent3Aux!=Entrada3)||(Ent 4Aux!=Entra
da4))
	{
		if (m_Socket1.GetState ()==7) EnviarEstadoAlSck();
	}
}

void CProyectoBDDlg::EnviarEstadoAlSck()
{
	CString Estado;

	Estado="Estado: ";

	if (Salida1==1)		Estado.Insert (9,"1 ");
	else Estado.Insert (9,"0 ");
	if (Salida2==1)		Estado.Insert (11,"1 ");
	else Estado.Insert (11,"0 ");
	if (Salida3==1)		Estado.Insert (13,"1 ");
	else Estado.Insert (13,"0 ");
	if (Salida4==1)		Estado.Insert (15,"1 ");
	else Estado.Insert (15,"0 ");
	if (Salida5==1)		Estado.Insert (17,"1 ");
	else Estado.Insert (17,"0 ");
	if (Salida6==1)		Estado.Insert (19,"1 ");
	else Estado.Insert (19,"0 ");
	if (Salida7==1)		Estado.Insert (21,"1 ");
	else Estado.Insert (21,"0 ");
	if (Salida8==1)		Estado.Insert (23,"1*");
	else Estado.Insert (23,"0*");
	if (Entrada1==1) Estado.Insert (25,"1 ");
	else Estado.Insert (25,"0 ");
	if (Entrada2==1) Estado.Insert (27,"1 ");
```

```
        else Estado.Insert (27,"0 ");
        if (Entrada3==1) Estado.Insert (29,"1 ");
        else Estado.Insert (29,"0 ");
        if (Entrada4==1) Estado.Insert (31,"1");
        else Estado.Insert (31,"0");

        //m_Socket1.ABREnviarSck (Estado);
        m_Socket1.ABREnviarSck (Estado);

}

void CProyectoBDDlg::EnviarLeyendaAlSck()
{
        CString Leyenda;
        CString Estado;

        Estado="Leyenda:";

        Leyenda.Insert (Leyenda.GetLength (),Leer(&m_EditSalida1));
        Leyenda.Insert (Leyenda.GetLength (),",");
        Leyenda.Insert (Leyenda.GetLength (),Leer(&m_EditSalida2));
        Leyenda.Insert (Leyenda.GetLength (),",");
        Leyenda.Insert (Leyenda.GetLength (),Leer(&m_EditSalida3));
        Leyenda.Insert (Leyenda.GetLength (),",");
        Leyenda.Insert (Leyenda.GetLength (),Leer(&m_EditSalida4));
        Leyenda.Insert (Leyenda.GetLength (),",");
        Leyenda.Insert (Leyenda.GetLength (),Leer(&m_EditSalida5));
        Leyenda.Insert (Leyenda.GetLength (),",");
        Leyenda.Insert (Leyenda.GetLength (),Leer(&m_EditSalida6));
        Leyenda.Insert (Leyenda.GetLength (),",");
        Leyenda.Insert (Leyenda.GetLength (),Leer(&m_EditSalida7));
        Leyenda.Insert (Leyenda.GetLength (),",");
        Leyenda.Insert (Leyenda.GetLength (),Leer(&m_EditSalida8));
        Leyenda.Insert (Leyenda.GetLength (),",");
        Leyenda.Insert (Leyenda.GetLength (),Leer(&m_EditEntrada1));
        Leyenda.Insert (Leyenda.GetLength (),",");
        Leyenda.Insert (Leyenda.GetLength (),Leer(&m_EditEntrada2));
        Leyenda.Insert (Leyenda.GetLength (),",");
        Leyenda.Insert (Leyenda.GetLength (),Leer(&m_EditEntrada3));
        Leyenda.Insert (Leyenda.GetLength (),",");
        Leyenda.Insert (Leyenda.GetLength (),Leer(&m_EditEntrada4));
        Leyenda.Insert (Leyenda.GetLength (),",");

        if (Salida1==1)       Estado.Insert (9,"1 ");
        else Estado.Insert (9,"0 ");
        if (Salida2==1)       Estado.Insert (11,"1 ");
        else Estado.Insert (11,"0 ");
        if (Salida3==1)       Estado.Insert (13,"1 ");
        else Estado.Insert (13,"0 ");
        if (Salida4==1)       Estado.Insert (15,"1 ");
        else Estado.Insert (15,"0 ");
        if (Salida5==1)       Estado.Insert (17,"1 ");
        else Estado.Insert (17,"0 ");
        if (Salida6==1)       Estado.Insert (19,"1 ");
        else Estado.Insert (19,"0 ");
        if (Salida7==1)       Estado.Insert (21,"1 ");
        else Estado.Insert (21,"0 ");
        if (Salida8==1)       Estado.Insert (23,"1*");
        else Estado.Insert (23,"0*");
        if (Entrada1==1) Estado.Insert (25,"1 ");
        else Estado.Insert (25,"0 ");
        if (Entrada2==1) Estado.Insert (27,"1 ");
        else Estado.Insert (27,"0 ");
        if (Entrada3==1) Estado.Insert (29,"1 ");
        else Estado.Insert (29,"0 ");
        if (Entrada4==1) Estado.Insert (31,"1");
        else Estado.Insert (31,"0");
        Estado.Insert (Estado.GetLength (),"*");
        Estado.Insert(Estado.GetLength (),Leyenda);

        m_Socket1.ABREnviarSck (Estado);
}

void CProyectoBDDlg::ImprimirLeyenda()
{
        CString aux;
```

```
        m_EditSalida1.SetWindowText (AfxGetApp()->GetProfileString
("Leyenda","Salida1"));
        m_EditSalida2.SetWindowText (AfxGetApp()->GetProfileString ("Leyenda","Salida2")
);
        m_EditSalida3.SetWindowText (AfxGetApp()->GetProfileString
("Leyenda","Salida3"));
        m_EditSalida4.SetWindowText (AfxGetApp()->GetProfileString
("Leyenda","Salida4"));
        m_EditSalida5.SetWindowText (AfxGetApp()->GetProfileString ("Leyenda","Salida5")
);
        m_EditSalida6.SetWindowText (AfxGetApp()->GetProfileString ("Leyenda","Salida6")
);
        m_EditSalida7.SetWindowText (AfxGetApp()->GetProfileString
("Leyenda","Salida7"));
        m_EditSalida8.SetWindowText (AfxGetApp()->GetProfileString ("Leyenda","Salida8")
);

        m_EditEntrada1.SetWindowText (AfxGetApp()->GetProfileString
("Leyenda","Entrada1"));
        m_EditEntrada2.SetWindowText (AfxGetApp()->GetProfileString
("Leyenda","Entrada2") );
        m_EditEntrada3.SetWindowText (AfxGetApp()->GetProfileString
("Leyenda","Entrada3") );
        m_EditEntrada4.SetWindowText (AfxGetApp()->GetProfileString
("Leyenda","Entrada4") );

        m_EditIP.SetWindowText (m_Socket1.GetLocalIP () );
        AfxGetApp()->WriteProfileString ("Socket","IP",m_Socket1.GetLocalIP ());

        m_EditPort.SetWindowText (AfxGetApp()->GetProfileString ("Socket","Port"));

        m_EditPath.SetWindowText (AfxGetApp()->GetProfileString ("BD","Path"));

        UpdateData(FALSE);

}
```

ABRWinsock.h, funciones útiles gestionar el control Winsock

```
//Se deben incluir las declaraciones de las funciones de este fichero
//en el fichero MSWinsockControl.h

void CMSWinsockControl::ABRConectarSck()
{
        // TODO: Add your control notification handler code here
        CString m_host;
        CString m_port;
        VARIANT vtHost;
        VARIANT vtPort;

//    if (AfxGetApp()->GetProfileString ("Socket2","IP")="")      AfxGetApp()-
>WriteProfileString ("Socket2","IP","127.0.0.1");
//      if (AfxGetApp()->GetProfileString ("Socket2","Port")="") AfxGetApp()-
>WriteProfileString ("Socket2","Port","888");

        m_host=AfxGetApp()->GetProfileString ("Socket2","IP");
        m_port=_T(AfxGetApp()->GetProfileString ("Socket2","Port"));

        //m_host="127.0.0.1";
        //m_port=_T("888");

        vtHost.vt=VT_BSTR;
        vtPort.vt=VT_BSTR;

         SetRemoteHost(m_host);

         SetRemotePort(atoi(m_port));

        vtHost.bstrVal=m_host.AllocSysString();
        vtPort.bstrVal=m_port.AllocSysString();

         Connect(vtHost,vtPort);


}

void CMSWinsockControl::ABROnConnectWinsock1()
{
        // TODO: Add your control notification handler code here

}

void CMSWinsockControl::ABREscucharSck()
{
        // TODO: Add your command handler code here
         Close ();
         Listen ();


}

void CMSWinsockControl::ABROnConnectionRequestWinsock1(long requestID)
{
        // TODO: Add your control notification handler code here
         Close ();
         Accept (requestID);


}

void CMSWinsockControl::ABROnDataArrivalWinsock1(long bytesTotal)
{
        // TODO: Add your control notification handler code here
        CString strport;
        VARIANT vtData;
        VARIANT vtType;
        VARIANT vtMaxlen;

        strport="4000";
        vtData.vt=VT_BSTR;
        vtType.vt=VT_ERROR;
```

```
        vtMaxlen.vt=VT_I2;
        vtMaxlen.iVal=(short)bytesTotal;


        CString Datos;


        vtData.bstrVal=strport.AllocSysString();
        vtType.bstrVal=strport.AllocSysString();


         GetData(&vtData,vtType,vtMaxlen);


        Datos=vtData.bstrVal;

        Dato=Datos;

}

void CMSWinsockControl::ABREnviarSck(CString datos)
{
        // TODO: Add your control notification handler code here
        VARIANT vtCommand;
        CString m_command;

        vtCommand.vt=VT_BSTR;
        m_command=datos;

        vtCommand.bstrVal =m_command.AllocSysString ();
         SendData (vtCommand);


}

void CMSWinsockControl::ABROnCloseWinsock1()
{
        // TODO: Add your control notification handler code here

}

void CMSWinsockControl::ABROnErrorWinsock1(short Number, BSTR FAR* Description, long
Scode, LPCTSTR Source, LPCTSTR HelpFile, long HelpContext, BOOL FAR* CancelDisplay)
{
        // TODO: Add your control notification handler code here

}

void CMSWinsockControl::ABRCerrarSck()
{
        // TODO: Add your command handler code here
         Close ();

}

void CMSWinsockControl::ABROnSendCompleteWinsock1 ()
{
        // TODO: Add your command handler code here

}


CString CMSWinsockControl::ABRConvertirEstado(int Estado)
{
        switch(Estado)
        {
    case 0:
                return "Closed";
         break;
    case 1 :
         return "Open";
         break;
    case 2 :
         return "Listening";
         break;
        case 3:
         return "Connection pendingOpen";
```

```
        break;
    case 4 :
        return "Resolving host";
        break;
    case 5 :
        return "Host resolved";
        break;
       case 6 :
        return "Connecting";
        break;
       case 7:
        return "Connected";
        break;
    case 8 :
        return "Peer is closing the connection";
        break;
    case 9 :
        return "Error";
        break;
       default:
              return "NULL";
              break;
       }

}
```

ABRCClienteDlgUtils.h, funciones útiles para el diálogo cliente

```
/*void CClienteDlg::InsertarRegistro(CString Nombre, CString Estado, CString Hora)
{
CString aux;

	m_pSet->m_Nombre=Nombre;
	m_pSet->m_Estado=Estado;
	m_pSet->m_Hora=Hora;
	EscribirRegistro();


	m_pSet=&m_BaseDatosDlgSet;
	m_pSet->ABRAbrirBD ();

	m_pSet->ABRInsertarRegistro(Nombre,Estado,Hora);

	m_pSet->ABRCerrarBD ();

}

CString CClienteDlg::InsertarHora()
{
	CString Hora;
	COleDateTime date=COleDateTime::GetCurrentTime ();

	Hora.Format ("%.2d/%.2d/%d
%.2d:%.2d:%.2d",date.GetDay(),date.GetMonth(),date.GetYear (),date.GetHour
(),date.GetMinute (),date.GetSecond ());

	return Hora;
}

void CClienteDlg::EscribirRegistro ()
{
	m_EditNombre.SetWindowText (m_pSet->m_Nombre);
	m_EditEstado.SetWindowText (m_pSet->m_Estado);
	m_EditHora.SetWindowText (m_pSet->m_Hora);
}
*/
void CClienteDlg::Escribir(CEdit* Edit, CString String)
{

	CString aux;

	Edit->GetWindowText (aux);
	aux.Insert (aux.GetLength (),"\r\n");
	aux.Insert (aux.GetLength (),String);
	Edit->SetWindowText (aux);
	Edit->SetSel (Edit->GetWindowTextLength (),Edit->GetWindowTextLength (),TRUE);
	Edit->LineScroll (Edit->GetScrollLimit(SB_VERT));

	UpdateData(FALSE);
}

CString CClienteDlg::Leer(CEdit *Edit)
{
	CString aux;
	Edit->GetWindowText (aux);
	return aux;
}

void CClienteDlg::InicializarSalidas()
{
	CString Estado;

	//Entrada1=PuertoParalelo.Entrada1 ();
	//Entrada2=PuertoParalelo.Entrada2 ();
	//Entrada3=PuertoParalelo.Entrada3 ();
	//Entrada4=PuertoParalelo.Entrada4 ();

	Estado=m_Socket2.Dato;

	if (Estado.Mid (8,1)=="1") Salida1=1;
```

```
        else Salida1=0;
        if (Estado.Mid (10,1)=="1") Salida2=1;
        else Salida2=0;
        if (Estado.Mid (12,1)=="1") Salida3=1;
        else Salida3=0;
        if (Estado.Mid (14,1)=="1") Salida4=1;
        else Salida4=0;
        if (Estado.Mid (16,1)=="1") Salida5=1;
        else Salida5=0;
        if (Estado.Mid (18,1)=="1") Salida6=1;
        else Salida6=0;
        if (Estado.Mid (20,1)=="1") Salida7=1;
        else Salida7=0;
        if (Estado.Mid (22,1)=="1") Salida8=1;
        else Salida8=0;

        if (Estado.Mid (24,1)=="1") Entrada1=1;
        else Entrada1=0;
        if (Estado.Mid (26,1)=="1") Entrada2=1;
        else Entrada2=0;
        if (Estado.Mid (28,1)=="1") Entrada3=1;
        else Entrada3=0;
        if (Estado.Mid (30,1)=="1") Entrada4=1;
        else Entrada4=0;

        ActualizarPuerto();

}

void CClienteDlg::InicializarLeyenda()
{
        CString Leyenda;
        CString Estado;
        CString aux;

        Leyenda=m_Socket2.Dato;

        Estado="Estado: ";
        Estado.Insert (Estado.GetLength (),Leyenda.Mid (8,23));
        Escribir(&m_EditRecibido,Estado);

        if (Estado.Mid (8,1)=="1") Salida1=1;
        else Salida1=0;
        if (Estado.Mid (10,1)=="1") Salida2=1;
        else Salida2=0;
        if (Estado.Mid (12,1)=="1") Salida3=1;
        else Salida3=0;
        if (Estado.Mid (14,1)=="1") Salida4=1;
        else Salida4=0;
        if (Estado.Mid (16,1)=="1") Salida5=1;
        else Salida5=0;
        if (Estado.Mid (18,1)=="1") Salida6=1;
        else Salida6=0;
        if (Estado.Mid (20,1)=="1") Salida7=1;
        else Salida7=0;
        if (Estado.Mid (22,1)=="1") Salida8=1;
        else Salida8=0;

        if (Estado.Mid (24,1)=="1") Entrada1=1;
        else Entrada1=0;
        if (Estado.Mid (26,1)=="1") Entrada2=1;
        else Entrada2=0;
        if (Estado.Mid (28,1)=="1") Entrada3=1;
        else Entrada3=0;
        if (Estado.Mid (30,1)=="1") Entrada4=1;
        else Entrada4=0;

        ActualizarPuerto();

        Leyenda=Leyenda.Mid (32,Leyenda.GetLength ());

        aux=Leyenda.Mid(0,Leyenda.FindOneOf (","));
        Leyenda=Leyenda.Mid(Leyenda.FindOneOf (",")+1,Leyenda.GetLength ());
        m_EditSalida1.SetWindowText(aux);
        aux=Leyenda.Mid(0,Leyenda.FindOneOf (","));
        Leyenda=Leyenda.Mid(Leyenda.FindOneOf (",")+1,Leyenda.GetLength ());
        m_EditSalida2.SetWindowText(aux);
```

```
        aux=Leyenda.Mid(0,Leyenda.FindOneOf (","));
        Leyenda=Leyenda.Mid(Leyenda.FindOneOf (",")+1,Leyenda.GetLength ());
        m_EditSalida3.SetWindowText(aux);
        aux=Leyenda.Mid(0,Leyenda.FindOneOf (","));
        Leyenda=Leyenda.Mid(Leyenda.FindOneOf (",")+1,Leyenda.GetLength ());
        m_EditSalida4.SetWindowText(aux);
        aux=Leyenda.Mid(0,Leyenda.FindOneOf (","));
        Leyenda=Leyenda.Mid(Leyenda.FindOneOf (",")+1,Leyenda.GetLength ());
        m_EditSalida5.SetWindowText(aux);
        aux=Leyenda.Mid(0,Leyenda.FindOneOf (","));
        Leyenda=Leyenda.Mid(Leyenda.FindOneOf (",")+1,Leyenda.GetLength ());
        m_EditSalida6.SetWindowText(aux);
        aux=Leyenda.Mid(0,Leyenda.FindOneOf (","));
        Leyenda=Leyenda.Mid(Leyenda.FindOne Of (",")+1,Leyenda.GetLength ());
        m_EditSalida7.SetWindowText(aux);
        aux=Leyenda.Mid(0,Leyenda.FindOneOf (","));
        Leyenda=Leyenda.Mid(Leyenda.FindOneOf (",")+1,Leyenda.GetLength ());
        m_EditSalida8.SetWindowText(aux);
        aux=Leyenda.Mid(0,Leyenda.FindOneOf (","));
        Leyenda=Leyenda.Mid(Leyenda.FindOneOf (",")+1,Leyenda.GetLength ());
        m_EditEntrada1.SetWindowText(aux);
        aux=Leyenda.Mid(0,Leyenda.FindOneOf (","));
        Leyenda=Leyenda.Mid(Leyenda.FindOneOf (",")+1,Leyenda.GetLength ());
        m_EditEntrada2.SetWindowText(aux);
        aux=Leyenda.Mid(0,Leyenda.FindOneOf (","));
        Leyenda=Leyenda.Mid(Leyenda.FindOneOf (",")+1,Leyenda.GetLength ());
        m_EditEntrada3.SetWindowText(aux);
        aux=Leyenda.Mid(0,Leyenda.FindOneOf (","));
        Leyenda=Leyenda.Mid(Leyenda.FindOneOf (",")+1,Leye nda.GetLength ());
        m_EditEntrada4.SetWindowText(aux);
}

void CClienteDlg::ActualizarPuerto()
{
        if (Salida1==1) m_ChkSal1.SetCheck (1);
        else m_ChkSal1.SetCheck (0);
        if (Salida2==1) m_ChkSal2.SetCheck (1);
        else m_ChkSal2.SetCheck (0);
        if (Salida3==1) m_ChkSal3.SetCheck (1);
        else m_ChkSal3.SetCheck (0);
        if (Salida4==1) m_ChkSal4.SetCheck (1);
        else m_ChkSal4.SetCheck (0);
        if (Salida5==1) m_ChkSal5.SetCheck (1);
        else m_ChkSal5.SetCheck (0);
        if (Salida6==1)        m_ChkSal6.SetCheck (1);
        else m_ChkSal6.SetCheck (0);
        if (Salida7==1) m_ChkSal7.SetCheck (1);
        else m_ChkSal7.SetCheck (0);
        if (Salida8==1) m_ChkSal8.SetCheck (1);
        else m_ChkSal8.SetCheck (0);

        if (Entrada1==1) m_ChkEnt1.SetCheck (1);
        else m_ChkEnt1.SetCheck (0);
        if (Entrada2==1) m_ChkEnt2.SetCheck (1);
        else m_ChkEnt2.SetCheck (0);
        if (Entrada3==1) m_ChkEnt3.SetCheck (1);
        else m_ChkEnt3.SetCheck (0);
        if (Entrada4==1) m_ChkEnt4.SetCheck (1);
        else m_ChkEnt4.SetCheck (0);
}
/*
void CClienteDlg::GuardarSalidas()
{
        CString Estado;

        if (Salida1==1)        Estado.Insert (1,"1 ");
        else Estado.Insert (1,"0 ");
        if (Salida2==1)        Estado.Insert (3,"1 ");
        else Estado.Insert (3,"0 ");
        if (Salida3==1)        Estado.Insert (5,"1 ");
        else Estado.Insert (5,"0 ");
        if (Salida4==1)        Estado.Insert (7,"1 ");
        else Estado.Insert (7,"0 ");
        if (Salida5==1)        Estado.Insert (9,"1 ");
        else Estado.Insert (9,"0 ");
        if (Salida6==1)        Estado.Insert (11,"1 ");
        else Estado.Insert (11,"0 ");
```

```
	if (Salida7==1)		Estado.Insert (13,"1 ");
	else Estado.Insert (13,"0 ");
	if (Salida8==1)		Estado.Insert (15,"1*");
	else Estado.Insert (15,"0*");
	if (Entrada1==1) Estado.Insert (17,"1 ");
	else Estado.Insert (17,"0 ");
	if (Entrada2==1) Estado.Insert (19,"1 ");
	else Estado.Insert (19,"0 ");
	if (Entrada3==1) Estado.Insert (21,"1 ");
	else Estado.Insert (21,"0 ");
	if (Entrada4==1) Estado.Insert (23,"1");
	else Estado.Insert (23,"0");

	m_EditNombre.SetWindowText ("Se Cierra Dialogo de Salidas");
	m_EditEstado.SetWindowText (Estado);
	m_EditHora.SetWindowText (InsertarHora());

	m_pSet-
>ABRInsertarRegistro(Leer(&m_EditNombre),Leer(&m_EditEstado),Leer(&m_EditHora));

}


CString CClienteDlg::MirarEstado(int nSal)
{
	CString out;

	switch (nSal)
	{
		case 1:
			if (Salida1) out="0";
			else out="1";
			Salida1=!Salida1;
			break;
		case 2:
			if (Salida2) out="0";
			else out="1";
			Salida2=!Salida2;
			break;
		case 3:
			if (Salida3) out="0";
			else out="1";
			Salida3=!Salida3;
			break;
		case 4:
			if (Salida4) out="0";
			else out="1";
			Salida4=!Salida4;
			break;
		case 5:
			if (Salida5) out="0";
			else out="1";
			Salida5=!Salida5;
			break;
		case 6:
			if (Salida6) out="0";
			else out="1";
			Salida6=!Salida6;
			break;
		case 7:
			if (Salida7) out="0";
			else out="1";
			Salida7=!Salida7;
			break;
		case 8:
			if (Salida8) out="0";
			else out="1";
			Salida8=!Salida8;
			break;
		case 9:
			if (!Entrada1) out="0";
			else out="1";
			break;
		case 10:
			if (!Entrada2) out="0";
			else out="1";
			break;
```

```
		case 11:
			if (!Entrada3) out="0";
			else out="1";
			break;
		case 12:
			if (!Entrada4) out="0";
			else out="1";
			break;
		default:
			return "NULL";

	}
	return out;
}



void CClienteDlg::LeerEntradasPuerto()
{
	int Ent1Aux;
	int Ent2Aux;
	int Ent3Aux;
	int Ent4Aux;
	//Guardar Entradas anteriores
	Ent1Aux=Entrada1;
	Ent2Aux=Entrada2;
	Ent3Aux=Entrada3;
	Ent4Aux=Entrada4;
	//Actualizar las entradas
	Entrada1=PuertoParalelo.Entrada1 ();
	Entrada2=PuertoParalelo.Entrada2 ();
	Entrada3=PuertoParalelo.Entrada3 ();
	Entrada4=PuertoParalelo.Entrada4 ();
	//Actualizar Entradas en el dialogo
	if (Entrada1==1) m_CheckEnt1.SetCheck (1);
	else m_CheckEnt1.SetCheck (0);
	if (Entrada2==1) m_CheckEnt2.SetCheck (1);
	else m_CheckEnt2.SetCheck (0);
	if (Entrada3==1) m_CheckEnt3.SetCheck (1);
	else m_CheckEnt3.SetCheck (0);
	if (Entrada4==1) m_CheckEnt4.SetCheck (1);
	else m_CheckEnt4.SetCheck (0);
	//Registro en la base de datos de los cambios
	if (Ent1Aux!=Entrada1)
	{
		InsertarRegistro("Entrada1",MirarEstado(9),InsertarHora());

	}
	if (Ent2Aux!=Entrada2)
	{
		InsertarRegistro("Entrada2",MirarEstado(10),InsertarHora());

	}
	if (Ent3Aux!=Entrada3)
	{
		InsertarRegistro("Entrada3",MirarEstado(11),InsertarHora());

	}
	if (Ent4Aux!=Entrada4)
	{
		InsertarRegistro("Entrada4",MirarEstado(12),InsertarHora());

	}
	// Enviar Estado al socket si hay cambios
	if((Ent1Aux!=Entrada1)||(Ent2Aux!=Entrada2)||(Ent3Aux!=Entrada3)||(Ent4Aux!=Entra
da4))
	{
		if (m_Socket1.GetState ()==7) EnviarEstadoAlSck();
	}
}

void CClienteDlg::EnviarEstadoAlSck()
{
	CString Estado;

	Estado="Estado: ";
```

```
        if (Salida1==1)       Estado.Insert (9,"1 ");
        else Estado.Insert (9,"0 ");
        if (Salida2==1)       Estado.Insert (11,"1 ");
        else Estado.Insert (11,"0 ");
        if (Salida3==1)       Estado.Insert (13,"1 ");
        else Estado.Insert (13,"0 ");
        if (Salida4==1)       Estado.Insert (15,"1 ");
        else Estado.Insert (15,"0 ");
        if (Salida5==1)       Estado.Insert (17,"1 ");
        else Estado.Insert (17,"0 ");
        if (Salida6==1)       Estado.Insert (19,"1 ");
        else Estado.Insert (19,"0 ");
        if (Salida7==1)       Estado.Insert (21,"1 ");
        else Estado.Insert (21,"0 ");
        if (Salida8==1)       Estado.Insert (23,"1*");
        else Estado.Insert (23,"0*");
        if (Entrada1==1) Estado.Insert (25,"1 ");
        else Estado.Insert (25,"0 ");
        if (Entrada2==1) Estado.Insert (27,"1 ");
        else Estado.Insert (27,"0 ");
        if (Entrada3==1) Estado.Insert (29,"1 ");
        else Estado.Insert (29,"0 ");
        if (Entrada4==1) Estado.Insert (31,"1");
        else Estado.Insert (31,"0");

        m_Socket1.ABREnviarSck (Estado);
}

void CClienteDlg::ImprimirLeyenda()
{
        CString aux;
        m_EditSalida1.SetWindowText (AfxGetApp()->GetProfileString
("Leyenda","Salida1"));
        m_EditSalida2.SetWindowText (AfxGetApp()->GetProfileString ("Leyenda","Salida2")
);
        m_EditSalida3.SetWindowText (AfxGetApp()->GetProfileString
("Leyenda","Salida3"));
        m_EditSalida4.SetWindowText (AfxGetApp()->GetProfileString
("Leyenda","Salida4"));
        m_EditSalida5.SetWindowText (AfxGetApp()->GetProfileString ("Leyenda","Salida5")
);
        m_EditSalida6.SetWindowText (AfxGetApp()->GetProfileString ("Leyenda","Salida6")
);
        m_EditSalida7.SetWindowText (AfxGetApp()->GetProfileString
("Leyenda","Salida7"));
        m_EditSalida8.SetWindowText (AfxGetApp()->GetProfileString ("Leyenda","Salida8")
);

        m_EditEntrada1.SetWindowText (AfxGetApp()->GetProfileString
("Leyenda","Entrada1"));
        m_EditEntrada2.SetWindowText (AfxGetApp()->GetProfileString
("Leyenda","Entrada2") );
        m_EditEntrada3.SetWindowText (AfxGetApp()->GetProfileString
("Leyenda","Entrada3") );
        m_EditEntrada4.SetWindowText (AfxGetApp()->GetProfileString
("Leyenda","Entrada4") );

        m_EditIP.SetWindowText (m_Socket1.GetLocalIP () );
        AfxGetApp()->WriteProfileString ("Socket","IP",m_Socket1.GetLocalIP ());

        m_EditPort.SetWindowText (AfxGetApp()->GetProfileString ("Socket","Port"));

        m_EditPath.SetWindowText (AfxGetApp()->GetProfileString ("BD","Path"));

        UpdateData(FALSE);

}*/
```

Instalación de la aplicación ProyectoBD.exe

Para la utilización de la aplicación de este proyecto en un ordenador PC compatible son necesarios los siguientes requisitos:

- Sistema Operativo Windows 95 o superior.
- Tarjeta de Puerto Paralelo en el PC.
- Tarjeta de red o modem para conexión a red TCP/IP.
- Control mswinsockcontrol.ocx registrado en el PC.
- Controlador ODBC de Microsoft Data Access instalado.

Con el disco de instalación se incluye el fichero de instalación de la aplicación, *ProyectoBDsetup.exe*.

Este fichero instalará el fichero de la aplicación *ProyectoBD.exe* y la base de datos necesaria para su utilización *bd1.mdb*.

Ahora la aplicación ya esta lista para se ejecutada en el ordenador instalado.
Menu Inicio -> Programas -> ABRProyectoBD -> ProyectoBD

La instalación se realiza de la misma manera para el PC servidor y para el PC cliente.

Si una vez instalado el programa este no funcionase correctamente es que falta alguno de los requisitos mencionados anteriormente. En el disco de instalación se han añadido todos los ficheros necesarios para que el programa pueda ejecutarse en el sistema donde falten alguno de los componentes necesarios.

- Para registrar el control, copiar el fichero mswinsockcontrol.ocx en *c:\windows\system32* y ejecutar:
  regsvr32.exe c:\windows\system32\mswinsockcontrol.ocx

- Para instalar el controlador ODBC de Microsoft Data Access ejecutar el fichero *Mdac_es.exe*.
  Esto instalará el fichero Mdac_typ.exe en español en *c:\Ipdk\Spanish\Mdac_typ.exe*, ejecutarlo y seguir los pasos de la instalación, este fichero añade las librerias necesarias para ODBC.

- Para instalar el *Mdac_es.exe* puede ser necesario tener instalado el fichero *DCOM98.exe* que carga controladores para dar soporte de comunicación entre los controles ActiveX.

Proyecto Final de Carrera:

CONTROL REMOTO MEDIANTE COMUNICACIÓN TCP/IP DE RADIOAYUDAS AERONAUTICAS

3.MANUAL DE USUARIO

- *Ingeniero Superior en Automática y Electrónica Industrial* — Agustín Baubí Rovira
- *Profesor Ponente* — Ernest Gil Dolcet

Febrero 2002

ESCOLA TÈCNICA SUPERIOR ENGINYERIA

UNIVERSITAT ROVIRA I VIRGILI

INDICE

Instalación de la aplicación ProyectoBD.exe 1
Funcionamiento de la aplicación 2
Dialogo principal. Aplicación servidor 3
Diálogo Base de Datos 4
Diálogo Propiedades 4
Diálogo Cliente 5

Proyecto Final de Carrera:

CONTROL REMOTO MEDIANTE COMUNICACIÓN TCP/IP DE RADIOAYUDAS AERONAUTICAS

MANUAL DE USUARIO

- *Ingeniero Superior*
  en Automática y Electrónica Industrial

Agustín Baubí Rovira

Febrero 2002

Instalación de la aplicación ProyectoBD.exe

Para la utilización de la aplicación de este proyecto en un ordenador PC compatible son necesarios los siguientes requisitos:

- Sistema Operativo Windows 95 o superior.
- Tarjeta de Puerto Paralelo en el PC.
- Tarjeta de red o modem para conexión a red TCP/IP.
- Control mswinsockcontrol.ocx registrado en el PC.
- Controlador ODBC de Microsoft Data Access instalado.

Con el disco de instalación se incluye el fichero de instalación de la aplicación, ProyectoBDsetup.exe.

Este fichero instalará el fichero de la aplicación ProyectoBD.exe y la base de datos necesaria para su utilización bd1.mdb.

Ahora la aplicación ya esta lista para se ejecutada en el ordenador instalado.
Menu Inicio -> Programas -> ABRProyectoBD -> ProyectoBD

La instalación se realiza de la misma manera para el PC servidor y para el PC cliente.

Si una vez instalado el programa este no funcionase correctamente es que falta alguno de los requisitos mencionados anteriormente. En el disco de instalación se han añadido todos los ficheros necesarios para que el programa pueda ejecutarse en el sistema donde falten alguno de los componentes necesarios.

- Para registrar el control, copiar el fichero mswinsockcontrol.ocx en c:\windows\system32 y ejecutar:
  regsvr32.exe c:\windows\system32\mswinsockcontrol.ocx
- Para instalar el controlador ODBC de Microsoft Data Access ejecutar el fichero Mdac_es.exe.
  Esto instalará el fichero Mdac_typ.exe en español en c:\Ipdk\Spanish\Mdac_typ.exe, ejecutarlo y seguir los pasos de la instalación, este fichero añade las librerias necesarias para ODBC.
- Para instalar el Mdac_es.exe puede ser necesario tener instalado el fichero DCOM98.exe que carga controladores para dar soporte de comunicación entre los controles ActiveX.

Funcionamiento de la aplicación

La aplicación creada (ProyectoBD.exe) se encarga de controlar ocho salidas y cuatro entradas digitales del puerto paralelo, realizar una base de datos con la información de los eventos que se han producido y funcionar como una aplicación servidor para establecer una comunicación con otra aplicación igual instalada en un ordenador conectado en la misma red.

El programa esta formado por cuatro diálogos:

- El diálogo principal. Aplicación servidor. Que se encarga de controlar el puerto paralelo y de realizar las funciones de servidor
- El dialogo de Base de datos. Que gestiona la base de datos donde se registran todos los eventos
- El dialogo Propiedades. Donde se configura la conexión y la leyenda de las entradas y salidas
- El diálogo Cliente. Que realiza la función de cliente para conectarse con el diálogo servidor en la dirección IP y Puerto que le asignemos

Con el programa incluimos una base de datos realizada con Microsoft Access para registrar los eventos que se producen. *(bd1.mdb)*

La base de datos dispone de una tabla llamada historico formada por tres campos de texto:

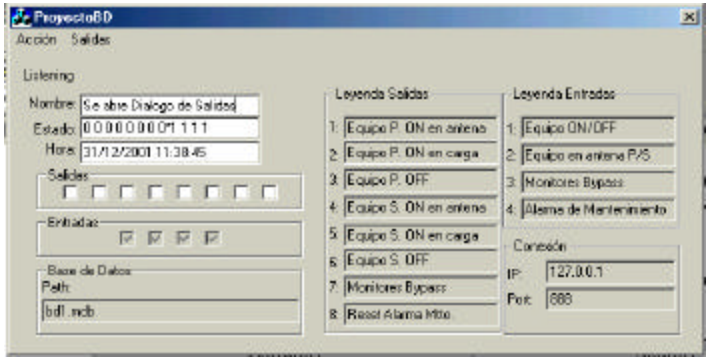
- Nombre, donde se registra el nombre del evento que se ha producido.
- Estado, donde se registra el estado final del evento que se ha producido.
- Hora, donde se registra la hora en que se ha producido el evento.

La aplicación realizada dispone de las funciones servidor y cliente, integrado todo en un mismo programa.

El programa servidor se ejecuta desde la computadora que se quiere controlar el puerto paralelo, pero también se instala en la computadora que se utiliza como cliente, aunque en este caso no se utilizan sus funciones de control del puerto y sólo se lanza el diálogo cliente para que la aplicación funcione como cliente del servidor remoto.

De esta manera una única aplicación se puede instalar en todos los ordenadores que se quiera y se pueden utilizar todos tanto como servidores como clientes.

Dialogo principal. Aplicación servidor



Al arrancar la aplicación la primera funciones que se ejecutan son la siguientes:

- Se abre la base de datos
- Se inicializan las salidas con los últimos estados registrados en la base de datos.
- Se inserta un registro en la base de datos con la información de apertura de la aplicación.
- Se cierra la base de datos.
- Se captura del registro de Windows la información de la leyenda que aparece en el diálogo.
- Cada 500 ms se comprueba si se ha producido algún cambio en las entradas digitales del puerto.
- Se pone el diálogo en modo de escuchar en la dirección de puerto que se indica en el diálogo.

Ahora la aplicación queda a la espera de que el usuario realice alguna acción sobre el dialogo del proyecto, sobre alguna opción del menú o se produzca algún cambio en las entradas del puerto, que se detecta automáticamente con la acción OnTimer()

- Si se actúa sobre alguna de las salidas o alguna opción del menú de salidas

  Se inserta un registro en la base de datos con la información de la salida activada y se actualiza el puerto paralelo con la nueva información. Si el servidor estuviera conectado a algún cliente le enviaría el nuevo estado de todas las entradas y salidas.

- Si se actúa con la opción Base de Datos del menú Acción.

  Se abre el diálogo de gestión de la Base de Datos de modo no modal (Se puede acceder a diálogo padre sin cerrar el propio diálogo)

- Si se actúa con la opción Propiedades del menú Acción

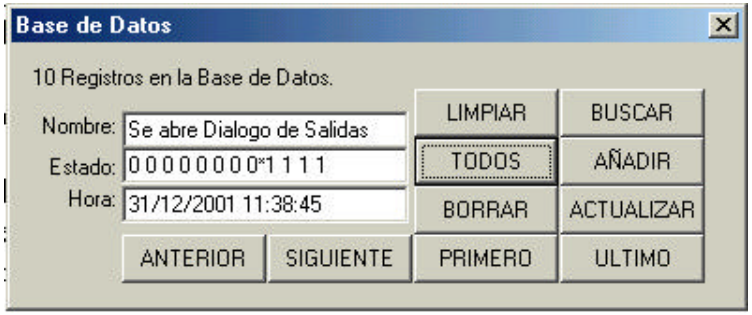
  Se abre el diálogo de modificación de propiedades del proyecto de modo modal (No se puede acceder a diálogo padre sin cerrar el propio diálogo).En el caso de realizar alguna modificación, se actualizan los datos del diálogo principal

- Si se actúa con la opción Cliente del menú Acción

  Se abre el diálogo cliente que funciona de modo modal, que funciona como cliente de cualquier otra aplicación servidor conectada a la red.

- Al cerrar la aplicación se registra el estado de las entradas y salidas y cierra la base de datos.

Diálogo Base de Datos



- Al arrancar el diálogo se abre la base de datos y se buscan todos los registros que contiene
- Luego se escribe el último registro en el diálogo

Ahora la aplicación queda a la espera de que el usuario realice alguna acción sobre el dialogo del Propiedades.

- Si se actúa sobre algún Botón del diálogo se realiza la función que indica.
- Si se modifica el contenido de algún registro se puede actualizar utilizando el botón que corresponde después de la modificación.

En cualquier momento se puede regresar al diálogo principal dejando es diálogo abierto o cerrarse como cualquier ventana de Windows

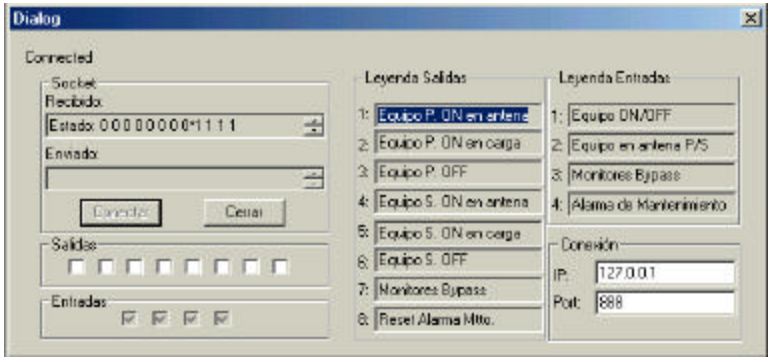
Diálogo Propiedades

- Al arrancar el diálogo se captura del registro de Windows la información de la leyenda que aparece en el diálogo

Ahora la aplicación queda a la espera de que el usuario realice alguna acción sobre el dialogo del Propiedades.

- Se puede modificar el contenido de los cuadros de texto y al pulsar el Botón OK escribe en el registro de Windows la información modificada y se regresa al diálogo principal
- Si cerrara el diálogo sin pulsar Ok no se registrarían las modificaciones realizadas en el registro de Windows

Diálogo Cliente



Al arrancar el diálogo se captura del registro de Windows la información de la conexión, IP, Port, que aparece en el diálogo

Ahora la aplicación queda a la espera de que el usuario realice alguna acción sobre el dialogo del Propiedades.

- Se puede modificar el contenido de los cuadros de texto IP y Port para indicar en que dirección y en que puerto se debe realizar la conexión con el servidor. Al realizar cualquier cambio se registran en el registro de Windows los nuevos datos actualizados

- Si se pulsa el Botón Conectar se realiza la conexión con el diálogo servidor.

- Al producirse la conexión la aplicación cliente empieza a recibir datos automáticamente, se recibe el estado de las salidas y entradas en el diálogo cliente y se inicializa la leyenda con la misma leyenda que tiene el diálogo servidor.

- Cada vez que se produce un cambio del estado de las entradas o salidas en el dialogo servidor y este esta conectado a un diálogo cliente, se recibe el nuevo estado actualizado.

- Si en el dialogo cliente se cambia el estado de alguna de las salidas se envía al servidor una orden de cambio de la salida correspondiente en el servidor y queda registrada en la base de datos del servidor como si se hubiera ejecutado desde él mismo.

- Si durante la conexión se pulsa el Botón Cerrar se cierra la conexión del cliente con el servidor y se puede volver a conectar otra vez con el mismo servidor o con otro diferente en otra IP y puerto distinto

- Si se cierra el diálogo también se cierra la conexión y se regresa al diálogo principal.

Proyecto Final de Carrera:

CONTROL REMOTO MEDIANTE COMUNICACIÓN TCP/IP DE RADIOAYUDAS AERONAUTICAS

4.ANEXOS

- *Ingeniero Superior en Automática y Electrónica Industrial* — Agustín Baubí Rovira
- *Profesor Ponente* — Ernest Gil Dolcet

Febrero 2002

```
1  CString CBaseDatosDlg::InsertarHora()
2  {
3      CString Hora;
4      COleDateTime date=COleDateTime::GetCurrentTime ();
5
6      Hora.Format ("%.2d/%.2d/%d %.2d:%.2d:%.2d",date.GetDay(),date.GetMonth(),date.GetYear (),dat
7
8      return Hora;
9  }
10
11 void CBaseDatosDlg::Escribir(CEdit *Edit, CString String)
12 {
13     CString aux;
14
15     Edit->GetWindowText (aux);
16     aux.Insert (aux.GetLength (),String);
17     aux.Insert (aux.GetLength (),"\r\n");
18     Edit->SetWindowText (aux);
19     Edit->SetSel (Edit->GetWindowTextLength (),Edit->GetWindowTextLength (),TRUE);
20     Edit->LineScroll (Edit->GetScrollLimit(SB_VERT));
21
22     UpdateData(FALSE);
23 }
24
25 CString CBaseDatosDlg::Leer(CEdit *Edit)
26 {
27     CString aux;
28     Edit->GetWindowText (aux);
29     return aux;
30 }
31
32 void CBaseDatosDlg::EscribirRegistro ()
33 {
34     m_EditNombre.SetWindowText (m_pSet->m_Nombre);
35     m_EditEstado.SetWindowText (m_pSet->m_Estado);
36     m_EditHora.SetWindowText (m_pSet->m_Hora);
37 }
38
39 void CBaseDatosDlg::BorrarCampos()
40 {
41     m_EditNombre.SetWindowText ("");
42     m_EditEstado.SetWindowText ("");
43     m_EditHora.SetWindowText ("");
44 }
45
```

```
/*void CClienteDlg::InsertarRegistro(CString Nombre, CString Estado, CString Hora)
{
CString aux;

    m_pSet->m_Nombre=Nombre;
    m_pSet->m_Estado=Estado;
    m_pSet->m_Hora=Hora;
    EscribirRegistro();


    m_pSet=&m_BaseDatosDlgSet;
    m_pSet->ABRAbrirBD ();

    m_pSet->ABRInsertarRegistro(Nombre,Estado,Hora);

    m_pSet->ABRCerrarBD ();

}

CString CClienteDlg::InsertarHora()
{
    CString Hora;
    COleDateTime date=COleDateTime::GetCurrentTime ();

    Hora.Format ("%.2d/%.2d/%d %.2d:%.2d:%.2d",date.GetDay(),date.GetMonth(),date.GetYear (),da

    return Hora;
}

void CClienteDlg::EscribirRegistro ()
{
    m_EditNombre.SetWindowText (m_pSet->m_Nombre);
    m_EditEstado.SetWindowText (m_pSet->m_Estado);
    m_EditHora.SetWindowText (m_pSet->m_Hora);
}
*/
void CClienteDlg::Escribir(CEdit* Edit, CString String)
{

    CString aux;

    Edit->GetWindowText (aux);
    aux.Insert (aux.GetLength (),"\r\n");
    aux.Insert (aux.GetLength (),String);
    Edit->SetWindowText (aux);
    Edit->SetSel (Edit->GetWindowTextLength (),Edit->GetWindowTextLength (),TRUE);
    Edit->LineScroll (Edit->GetScrollLimit(SB_VERT));

    UpdateData(FALSE);
}

CString CClienteDlg::Leer(CEdit *Edit)
{
    CString aux;
    Edit->GetWindowText (aux);
    return aux;
}

void CClienteDlg::InicializarSalidas()
{
    CString Estado;

    //Entrada1=PuertoParalelo.Entrada1 ();
    //Entrada2=PuertoParalelo.Entrada2 ();
```

```
65      //Entrada3=PuertoParalelo.Entrada3 ();
66      //Entrada4=PuertoParalelo.Entrada4 ();
67
68      Estado=m_Socket2.Dato;
69
70      if (Estado.Mid (8,1)=="1") Salida1=1;
71      else Salida1=0;
72      if (Estado.Mid (10,1)=="1") Salida2=1;
73      else Salida2=0;
74      if (Estado.Mid (12,1)=="1") Salida3=1;
75      else Salida3=0;
76      if (Estado.Mid (14,1)=="1") Salida4=1;
77      else Salida4=0;
78      if (Estado.Mid (16,1)=="1") Salida5=1;
79      else Salida5=0;
80      if (Estado.Mid (18,1)=="1") Salida6=1;
81      else Salida6=0;
82      if (Estado.Mid (20,1)=="1") Salida7=1;
83      else Salida7=0;
84      if (Estado.Mid (22,1)=="1") Salida8=1;
85      else Salida8=0;
86
87      if (Estado.Mid (24,1)=="1") Entrada1=1;
88      else Entrada1=0;
89      if (Estado.Mid (26,1)=="1") Entrada2=1;
90      else Entrada2=0;
91      if (Estado.Mid (28,1)=="1") Entrada3=1;
92      else Entrada3=0;
93      if (Estado.Mid (30,1)=="1") Entrada4=1;
94      else Entrada4=0;
95
96      ActualizarPuerto();
97
98  }
99
100 void CClienteDlg::InicializarLeyenda()
101 {
102     CString Leyenda;
103     CString Estado;
104     CString aux;
105
106     Leyenda=m_Socket2.Dato;
107
108     Estado="Estado: ";
109     Estado.Insert (Estado.GetLength (),Leyenda.Mid (8,23));
110     Escribir(&m_EditRecibido,Estado);
111
112     if (Estado.Mid (8,1)=="1") Salida1=1;
113     else Salida1=0;
114     if (Estado.Mid (10,1)=="1") Salida2=1;
115     else Salida2=0;
116     if (Estado.Mid (12,1)=="1") Salida3=1;
117     else Salida3=0;
118     if (Estado.Mid (14,1)=="1") Salida4=1;
119     else Salida4=0;
120     if (Estado.Mid (16,1)=="1") Salida5=1;
121     else Salida5=0;
122     if (Estado.Mid (18,1)=="1") Salida6=1;
123     else Salida6=0;
124     if (Estado.Mid (20,1)=="1") Salida7=1;
125     else Salida7=0;
126     if (Estado.Mid (22,1)=="1") Salida8=1;
127     else Salida8=0;
128
```

```
129     if (Estado.Mid (24,1)=="1") Entrada1=1;
130     else Entrada1=0;
131     if (Estado.Mid (26,1)=="1") Entrada2=1;
132     else Entrada2=0;
133     if (Estado.Mid (28,1)=="1") Entrada3=1;
134     else Entrada3=0;
135     if (Estado.Mid (30,1)=="1") Entrada4=1;
136     else Entrada4=0;
137
138     ActualizarPuerto();
139
140     Leyenda=Leyenda.Mid (32,Leyenda.GetLength ());
141
142     aux=Leyenda.Mid(0,Leyenda.FindOneOf (","));
143     Leyenda=Leyenda.Mid(Leyenda.FindOneOf (",")+1,Leyenda.GetLength ());
144     m_EditSalida1.SetWindowText(aux);
145     aux=Leyenda.Mid(0,Leyenda.FindOneOf (","));
146     Leyenda=Leyenda.Mid(Leyenda.FindOneOf (",")+1,Leyenda.GetLength ());
147     m_EditSalida2.SetWindowText(aux);
148     aux=Leyenda.Mid(0,Leyenda.FindOneOf (","));
149     Leyenda=Leyenda.Mid(Leyenda.FindOneOf (",")+1,Leyenda.GetLength ());
150     m_EditSalida3.SetWindowText(aux);
151     aux=Leyenda.Mid(0,Leyenda.FindOneOf (","));
152     Leyenda=Leyenda.Mid(Leyenda.FindOneOf (",")+1,Leyenda.GetLength ());
153     m_EditSalida4.SetWindowText(aux);
154     aux=Leyenda.Mid(0,Leyenda.FindOneOf (","));
155     Leyenda=Leyenda.Mid(Leyenda.FindOneOf (",")+1,Leyenda.GetLength ());
156     m_EditSalida5.SetWindowText(aux);
157     aux=Leyenda.Mid(0,Leyenda.FindOneOf (","));
158     Leyenda=Leyenda.Mid(Leyenda.FindOneOf (",")+1,Leyenda.GetLength ());
159     m_EditSalida6.SetWindowText(aux);
160     aux=Leyenda.Mid(0,Leyenda.FindOneOf (","));
161     Leyenda=Leyenda.Mid(Leyenda.FindOneOf (",")+1,Leyenda.GetLength ());
162     m_EditSalida7.SetWindowText(aux);
163     aux=Leyenda.Mid(0,Leyenda.FindOneOf (","));
164     Leyenda=Leyenda.Mid(Leyenda.FindOneOf (",")+1,Leyenda.GetLength ());
165     m_EditSalida8.SetWindowText(aux);
166     aux=Leyenda.Mid(0,Leyenda.FindOneOf (","));
167     Leyenda=Leyenda.Mid(Leyenda.FindOneOf (",")+1,Leyenda.GetLength ());
168     m_EditEntrada1.SetWindowText(aux);
169     aux=Leyenda.Mid(0,Leyenda.FindOneOf (","));
170     Leyenda=Leyenda.Mid(Leyenda.FindOneOf (",")+1,Leyenda.GetLength ());
171     m_EditEntrada2.SetWindowText(aux);
172     aux=Leyenda.Mid(0,Leyenda.FindOneOf (","));
173     Leyenda=Leyenda.Mid(Leyenda.FindOneOf (",")+1,Leyenda.GetLength ());
174     m_EditEntrada3.SetWindowText(aux);
175     aux=Leyenda.Mid(0,Leyenda.FindOneOf (","));
176     Leyenda=Leyenda.Mid(Leyenda.FindOneOf (",")+1,Leyenda.GetLength ());
177     m_EditEntrada4.SetWindowText(aux);
178 }
179
180 void CClienteDlg::ActualizarPuerto()
181 {
182     if (Salida1==1) m_ChkSal1.SetCheck (1);
183     else m_ChkSal1.SetCheck (0);
184     if (Salida2==1) m_ChkSal2.SetCheck (1);
185     else m_ChkSal2.SetCheck (0);
186     if (Salida3==1) m_ChkSal3.SetCheck (1);
187     else m_ChkSal3.SetCheck (0);
188     if (Salida4==1) m_ChkSal4.SetCheck (1);
189     else m_ChkSal4.SetCheck (0);
190     if (Salida5==1) m_ChkSal5.SetCheck (1);
191     else m_ChkSal5.SetCheck (0);
192     if (Salida6==1) m_ChkSal6.SetCheck (1);
```

```
193     else m_ChkSal6.SetCheck (0);
194     if (Salida7==1) m_ChkSal7.SetCheck (1);
195     else m_ChkSal7.SetCheck (0);
196     if (Salida8==1) m_ChkSal8.SetCheck (1);
197     else m_ChkSal8.SetCheck (0);
198
199     if (Entrada1==1) m_ChkEnt1.SetCheck (1);
200     else m_ChkEnt1.SetCheck (0);
201     if (Entrada2==1) m_ChkEnt2.SetCheck (1);
202     else m_ChkEnt2.SetCheck (0);
203     if (Entrada3==1) m_ChkEnt3.SetCheck (1);
204     else m_ChkEnt3.SetCheck (0);
205     if (Entrada4==1) m_ChkEnt4.SetCheck (1);
206     else m_ChkEnt4.SetCheck (0);
207 }
208 /*
209 void CClienteDlg::GuardarSalidas()
210 {
211     CString Estado;
212
213     if (Salida1==1) Estado.Insert (1,"1 ");
214     else Estado.Insert (1,"0 ");
215     if (Salida2==1) Estado.Insert (3,"1 ");
216     else Estado.Insert (3,"0 ");
217     if (Salida3==1) Estado.Insert (5,"1 ");
218     else Estado.Insert (5,"0 ");
219     if (Salida4==1) Estado.Insert (7,"1 ");
220     else Estado.Insert (7,"0 ");
221     if (Salida5==1) Estado.Insert (9,"1 ");
222     else Estado.Insert (9,"0 ");
223     if (Salida6==1) Estado.Insert (11,"1 ");
224     else Estado.Insert (11,"0 ");
225     if (Salida7==1) Estado.Insert (13,"1 ");
226     else Estado.Insert (13,"0 ");
227     if (Salida8==1) Estado.Insert (15,"1*");
228     else Estado.Insert (15,"0*");
229     if (Entrada1==1) Estado.Insert (17,"1 ");
230     else Estado.Insert (17,"0 ");
231     if (Entrada2==1) Estado.Insert (19,"1 ");
232     else Estado.Insert (19,"0 ");
233     if (Entrada3==1) Estado.Insert (21,"1 ");
234     else Estado.Insert (21,"0 ");
235     if (Entrada4==1) Estado.Insert (23,"1");
236     else Estado.Insert (23,"0");
237
238     m_EditNombre.SetWindowText ("Se Cierra Dialogo de Salidas");
239     m_EditEstado.SetWindowText (Estado);
240     m_EditHora.SetWindowText (InsertarHora());
241
242     m_pSet->ABRInsertarRegistro(Leer(&m_EditNombre),Leer(&m_EditEstado),Leer(&m_EditHora));
243
244 }
245
246
247 CString CClienteDlg::MirarEstado(int nSal)
248 {
249     CString out;
250
251     switch (nSal)
252     {
253         case 1:
254             if (Salida1) out="0";
255             else out="1";
256             Salida1=!Salida1;
```

```
257             break;
258         case 2:
259             if (Salida2) out="0";
260             else out="1";
261             Salida2=!Salida2;
262             break;
263         case 3:
264             if (Salida3) out="0";
265             else out="1";
266             Salida3=!Salida3;
267             break;
268         case 4:
269             if (Salida4) out="0";
270             else out="1";
271             Salida4=!Salida4;
272             break;
273         case 5:
274             if (Salida5) out="0";
275             else out="1";
276             Salida5=!Salida5;
277             break;
278         case 6:
279             if (Salida6) out="0";
280             else out="1";
281             Salida6=!Salida6;
282             break;
283         case 7:
284             if (Salida7) out="0";
285             else out="1";
286             Salida7=!Salida7;
287             break;
288         case 8:
289             if (Salida8) out="0";
290             else out="1";
291             Salida8=!Salida8;
292             break;
293         case 9:
294             if (!Entrada1) out="0";
295             else out="1";
296             break;
297         case 10:
298             if (!Entrada2) out="0";
299             else out="1";
300             break;
301         case 11:
302             if (!Entrada3) out="0";
303             else out="1";
304             break;
305         case 12:
306             if (!Entrada4) out="0";
307             else out="1";
308             break;
309         default:
310             return "NULL";
311
312     }
313     return out;
314 }
315
316
317
318
319 void CClienteDlg::LeerEntradasPuerto()
320 {
```

```
321     int Ent1Aux;
322     int Ent2Aux;
323     int Ent3Aux;
324     int Ent4Aux;
325     //Guardar Entradas anteriores
326     Ent1Aux=Entrada1;
327     Ent2Aux=Entrada2;
328     Ent3Aux=Entrada3;
329     Ent4Aux=Entrada4;
330     //Actualizar las entradas
331     Entrada1=PuertoParalelo.Entrada1 ();
332     Entrada2=PuertoParalelo.Entrada2 ();
333     Entrada3=PuertoParalelo.Entrada3 ();
334     Entrada4=PuertoParalelo.Entrada4 ();
335     //Actualizar Entradas en el dialogo
336     if (Entrada1==1) m_CheckEnt1.SetCheck (1);
337     else m_CheckEnt1.SetCheck (0);
338     if (Entrada2==1) m_CheckEnt2.SetCheck (1);
339     else m_CheckEnt2.SetCheck (0);
340     if (Entrada3==1) m_CheckEnt3.SetCheck (1);
341     else m_CheckEnt3.SetCheck (0);
342     if (Entrada4==1) m_CheckEnt4.SetCheck (1);
343     else m_CheckEnt4.SetCheck (0);
344     //Registro en la base de datos de los cambios
345     if (Ent1Aux!=Entrada1)
346     {
347         InsertarRegistro("Entrada1",MirarEstado(9),InsertarHora());
348 
349     }
350     if (Ent2Aux!=Entrada2)
351     {
352         InsertarRegistro("Entrada2",MirarEstado(10),InsertarHora());
353 
354     }
355     if (Ent3Aux!=Entrada3)
356     {
357         InsertarRegistro("Entrada3",MirarEstado(11),InsertarHora());
358 
359     }
360     if (Ent4Aux!=Entrada4)
361     {
362         InsertarRegistro("Entrada4",MirarEstado(12),InsertarHora());
363 
364     }
365     // Enviar Estado al socket si hay cambios
366     if((Ent1Aux!=Entrada1)||(Ent2Aux!=Entrada2)||(Ent3Aux!=Entrada3)||(Ent4Aux!=Entrada4))
367     {
368         if (m_Socket1.GetState ()==7) EnviarEstadoAlSck();
369     }
370 }
371 
372 void CClienteDlg::EnviarEstadoAlSck()
373 {
374     CString Estado;
375 
376     Estado="Estado: ";
377 
378     if (Salida1==1) Estado.Insert (9,"1 ");
379     else Estado.Insert (9,"0 ");
380     if (Salida2==1) Estado.Insert (11,"1 ");
381     else Estado.Insert (11,"0 ");
382     if (Salida3==1) Estado.Insert (13,"1 ");
383     else Estado.Insert (13,"0 ");
384     if (Salida4==1) Estado.Insert (15,"1 ");
```

```
385     else Estado.Insert (15,"0 ");
386     if (Salida5==1) Estado.Insert (17,"1 ");
387     else Estado.Insert (17,"0 ");
388     if (Salida6==1) Estado.Insert (19,"1 ");
389     else Estado.Insert (19,"0 ");
390     if (Salida7==1) Estado.Insert (21,"1 ");
391     else Estado.Insert (21,"0 ");
392     if (Salida8==1) Estado.Insert (23,"1*");
393     else Estado.Insert (23,"0*");
394     if (Entrada1==1) Estado.Insert (25,"1 ");
395     else Estado.Insert (25,"0 ");
396     if (Entrada2==1) Estado.Insert (27,"1 ");
397     else Estado.Insert (27,"0 ");
398     if (Entrada3==1) Estado.Insert (29,"1 ");
399     else Estado.Insert (29,"0 ");
400     if (Entrada4==1) Estado.Insert (31,"1");
401     else Estado.Insert (31,"0");
402
403     m_Socket1.ABREnviarSck (Estado);
404 }
405
406 void CClienteDlg::ImprimirLeyenda()
407 {
408     CString aux;
409     m_EditSalida1.SetWindowText (AfxGetApp()->GetProfileString ("Leyenda","Salida1"));
410     m_EditSalida2.SetWindowText (AfxGetApp()->GetProfileString ("Leyenda","Salida2") );
411     m_EditSalida3.SetWindowText (AfxGetApp()->GetProfileString ("Leyenda","Salida3"));
412     m_EditSalida4.SetWindowText (AfxGetApp()->GetProfileString ("Leyenda","Salida4"));
413     m_EditSalida5.SetWindowText (AfxGetApp()->GetProfileString ("Leyenda","Salida5") );
414     m_EditSalida6.SetWindowText (AfxGetApp()->GetProfileString ("Leyenda","Salida6") );
415     m_EditSalida7.SetWindowText (AfxGetApp()->GetProfileString ("Leyenda","Salida7"));
416     m_EditSalida8.SetWindowText (AfxGetApp()->GetProfileString ("Leyenda","Salida8") );
417
418     m_EditEntrada1.SetWindowText (AfxGetApp()->GetProfileString ("Leyenda","Entrada1"));
419     m_EditEntrada2.SetWindowText (AfxGetApp()->GetProfileString ("Leyenda","Entrada2") );
420     m_EditEntrada3.SetWindowText (AfxGetApp()->GetProfileString ("Leyenda","Entrada3") );
421     m_EditEntrada4.SetWindowText (AfxGetApp()->GetProfileString ("Leyenda","Entrada4") );
422
423     m_EditIP.SetWindowText (m_Socket1.GetLocalIP () );
424     AfxGetApp()->WriteProfileString ("Socket","IP",m_Socket1.GetLocalIP ());
425
426     m_EditPort.SetWindowText (AfxGetApp()->GetProfileString ("Socket","Port"));
427
428     m_EditPath.SetWindowText (AfxGetApp()->GetProfileString ("BD","Path"));
429
430     UpdateData(FALSE);
431
432 }*/
```

```
1  void CProyectoBDDlg::InsertarRegistro(CString Nombre, CString Estado, CString Hora)
2  {
3  CString aux;
4
5      m_pSet->m_Nombre=Nombre;
6      m_pSet->m_Estado=Estado;
7      m_pSet->m_Hora=Hora;
8      EscribirRegistro();
9
10
11     m_pSet=&m_BaseDatosDlgSet;
12     m_pSet->ABRAbrirBD ();
13
14     m_pSet->ABRInsertarRegistro(Nombre,Estado,Hora);
15
16     m_pSet->ABRCerrarBD ();
17
18 }
19
20 CString CProyectoBDDlg::InsertarHora()
21 {
22     CString Hora;
23     COleDateTime date=COleDateTime::GetCurrentTime ();
24
25     Hora.Format ("%.2d/%.2d/%d %.2d:%.2d:%.2d",date.GetDay(),date.GetMonth(),date.GetYear (),da
26
27     return Hora;
28 }
29
30 void CProyectoBDDlg::EscribirRegistro ()
31 {
32     m_EditNombre.SetWindowText (m_pSet->m_Nombre);
33     m_EditEstado.SetWindowText (m_pSet->m_Estado);
34     m_EditHora.SetWindowText (m_pSet->m_Hora);
35 }
36
37 CString CProyectoBDDlg::Leer(CEdit *Edit)
38 {
39     CString aux;
40     Edit->GetWindowText (aux);
41     return aux;
42 }
43
44 void CProyectoBDDlg::InicializarSalidas()
45 {
46     CString Estado;
47
48     /*Entrada1=PuertoParalelo.Entrada1 ();
49     Entrada2=PuertoParalelo.Entrada2 ();
50     Entrada3=PuertoParalelo.Entrada3 ();
51     Entrada4=PuertoParalelo.Entrada4 ();
52 */
53     m_pSet->ABRBuscarRegistro ("Se Cierra Dialogo de Salidas","","");
54     if (m_pSet->m_Nombre!="Se Cierra Dialogo de Salidas")
55     {
56         m_pSet->ABRInsertarRegistro ("Inicio de la Base de Datos","-",InsertarHora());
57         Estado="0 0 0 0 0 0 0 0*0 0 0 0" ;
58     }
59     else
60     {
61         m_pSet->ABRUltimoRegistro();
62         Estado=m_pSet->m_Estado ;
63     }
64
```

```
65
66      if (Estado.Mid (0,1)=="1") Salida1=1;
67      else Salida1=0;
68      if (Estado.Mid (2,1)=="1") Salida2=1;
69      else Salida2=0;
70      if (Estado.Mid (4,1)=="1") Salida3=1;
71      else Salida3=0;
72      if (Estado.Mid (6,1)=="1") Salida4=1;
73      else Salida4=0;
74      if (Estado.Mid (8,1)=="1") Salida5=1;
75      else Salida5=0;
76      if (Estado.Mid (10,1)=="1") Salida6=1;
77      else Salida6=0;
78      if (Estado.Mid (12,1)=="1") Salida7=1;
79      else Salida7=0;
80      if (Estado.Mid (14,1)=="1") Salida8=1;
81      else Salida8=0;
82      if (Estado.Mid (16,1)=="1") Entrada1=1;
83      else Entrada1=0;
84      if (Estado.Mid (18,1)=="1") Entrada2=1;
85      else Entrada2=0;
86      if (Estado.Mid (20,1)=="1") Entrada3=1;
87      else Entrada3=0;
88      if (Estado.Mid (22,1)=="1") Entrada4=1;
89      else Entrada4=0;
90
91
92      m_EditNombre.SetWindowText ("Se abre Dialogo de Salidas");
93      m_EditEstado.SetWindowText (Estado);
94      m_EditHora.SetWindowText (InsertarHora());
95
96      InsertarRegistro(Leer(&m_EditNombre),Leer(&m_EditEstado),Leer(&m_EditHora));
97      ActualizarPuerto();
98  }
99
100 void CProyectoBDDlg::GuardarSalidas()
101 {
102     CString Estado;
103
104     if (Salida1==1) Estado.Insert (1,"1 ");
105     else Estado.Insert (1,"0 ");
106     if (Salida2==1) Estado.Insert (3,"1 ");
107     else Estado.Insert (3,"0 ");
108     if (Salida3==1) Estado.Insert (5,"1 ");
109     else Estado.Insert (5,"0 ");
110     if (Salida4==1) Estado.Insert (7,"1 ");
111     else Estado.Insert (7,"0 ");
112     if (Salida5==1) Estado.Insert (9,"1 ");
113     else Estado.Insert (9,"0 ");
114     if (Salida6==1) Estado.Insert (11,"1 ");
115     else Estado.Insert (11,"0 ");
116     if (Salida7==1) Estado.Insert (13,"1 ");
117     else Estado.Insert (13,"0 ");
118     if (Salida8==1) Estado.Insert (15,"1*");
119     else Estado.Insert (15,"0*");
120     if (Entrada1==1) Estado.Insert (17,"1 ");
121     else Estado.Insert (17,"0 ");
122     if (Entrada2==1) Estado.Insert (19,"1 ");
123     else Estado.Insert (19,"0 ");
124     if (Entrada3==1) Estado.Insert (21,"1 ");
125     else Estado.Insert (21,"0 ");
126     if (Entrada4==1) Estado.Insert (23,"1");
127     else Estado.Insert (23,"0");
128
```

```
129     m_EditNombre.SetWindowText ("Se Cierra Dialogo de Salidas");
130     m_EditEstado.SetWindowText (Estado);
131     m_EditHora.SetWindowText (InsertarHora());
132 
133     m_pSet->ABRInsertarRegistro(Leer(&m_EditNombre),Leer(&m_EditEstado),Leer(&m_EditHora));
134 
135 }
136 
137 
138 CString CProyectoBDDlg::MirarEstado(int nSal)
139 {
140     CString out;
141 
142     switch (nSal)
143     {
144         case 1:
145             if (Salida1) out="0";
146             else out="1";
147             Salida1=!Salida1;
148             break;
149         case 2:
150             if (Salida2) out="0";
151             else out="1";
152             Salida2=!Salida2;
153             break;
154         case 3:
155             if (Salida3) out="0";
156             else out="1";
157             Salida3=!Salida3;
158             break;
159         case 4:
160             if (Salida4) out="0";
161             else out="1";
162             Salida4=!Salida4;
163             break;
164         case 5:
165             if (Salida5) out="0";
166             else out="1";
167             Salida5=!Salida5;
168             break;
169         case 6:
170             if (Salida6) out="0";
171             else out="1";
172             Salida6=!Salida6;
173             break;
174         case 7:
175             if (Salida7) out="0";
176             else out="1";
177             Salida7=!Salida7;
178             break;
179         case 8:
180             if (Salida8) out="0";
181             else out="1";
182             Salida8=!Salida8;
183             break;
184         case 9:
185             if (!Entrada1) out="0";
186             else out="1";
187             break;
188         case 10:
189             if (!Entrada2) out="0";
190             else out="1";
191             break;
192         case 11:
```

```
193             if (!Entrada3) out="0";
194             else out="1";
195             break;
196         case 12:
197             if (!Entrada4) out="0";
198             else out="1";
199             break;
200         default:
201             return "NULL";
202 
203     }
204     return out;
205 }
206 
207 
208 void CProyectoBDDlg::ActualizarPuerto()
209 {
210     if (Salida1==1)
211     {
212         PuertoParalelo.Salida1 (1);
213         m_CheckSal1.SetCheck (1);
214     }
215     else
216     {
217         PuertoParalelo.Salida1 (0);
218         m_CheckSal1.SetCheck (0);
219     }
220     if (Salida2==1)
221     {
222         PuertoParalelo.Salida2 (1);
223         m_CheckSal2.SetCheck (1);
224     }
225     else
226     {
227         PuertoParalelo.Salida2 (0);
228         m_CheckSal2.SetCheck (0);
229     }
230     if (Salida3==1)
231     {
232         PuertoParalelo.Salida3 (1);
233         m_CheckSal3.SetCheck (1);
234     }
235     else
236     {
237         PuertoParalelo.Salida3 (0);
238         m_CheckSal3.SetCheck (0);
239     }
240     if (Salida4==1)
241     {
242         PuertoParalelo.Salida4 (1);
243         m_CheckSal4.SetCheck (1);
244     }
245     else
246     {
247         PuertoParalelo.Salida4 (0);
248         m_CheckSal4.SetCheck (0);
249     }
250     if (Salida5==1)
251     {
252         PuertoParalelo.Salida5 (1);
253         m_CheckSal5.SetCheck (1);
254     }
255     else
256     {
```

```
257         PuertoParalelo.Salida5 (0);
258         m_CheckSal5.SetCheck (0);
259     }
260     if (Salida6==1)
261     {
262         PuertoParalelo.Salida6 (1);
263         m_CheckSal6.SetCheck (1);
264     }
265     else
266     {
267         PuertoParalelo.Salida6 (0);
268         m_CheckSal6.SetCheck (0);
269     }
270     if (Salida7==1)
271     {
272         PuertoParalelo.Salida7 (1);
273         m_CheckSal7.SetCheck (1);
274     }
275     else
276     {
277         PuertoParalelo.Salida7 (0);
278         m_CheckSal7.SetCheck (0);
279     }
280     if (Salida8==1)
281     {
282         PuertoParalelo.Salida8 (1);
283         m_CheckSal8.SetCheck (1);
284     }
285     else
286     {
287         PuertoParalelo.Salida8 (0);
288         m_CheckSal8.SetCheck (0);
289     }
290
291     // Enviar Nuevo Estado al Socket
292
293     if (m_Socket1.GetState ()==7) EnviarEstadoAlSck();
294
295 }
296
297 void CProyectoBDDlg::LeerEntradasPuerto()
298 {
299     int Ent1Aux;
300     int Ent2Aux;
301     int Ent3Aux;
302     int Ent4Aux;
303     //Guardar Entradas anteriores
304     Ent1Aux=Entrada1;
305     Ent2Aux=Entrada2;
306     Ent3Aux=Entrada3;
307     Ent4Aux=Entrada4;
308     //Actualizar las entradas
309     Entrada1=PuertoParalelo.Entrada1 ();
310     Entrada2=PuertoParalelo.Entrada2 ();
311     Entrada3=PuertoParalelo.Entrada3 ();
312     Entrada4=PuertoParalelo.Entrada4 ();
313     //Actualizar Entradas en el dialogo
314     if (Entrada1==1) m_CheckEnt1.SetCheck (1);
315     else m_CheckEnt1.SetCheck (0);
316     if (Entrada2==1) m_CheckEnt2.SetCheck (1);
317     else m_CheckEnt2.SetCheck (0);
318     if (Entrada3==1) m_CheckEnt3.SetCheck (1);
319     else m_CheckEnt3.SetCheck (0);
320     if (Entrada4==1) m_CheckEnt4.SetCheck (1);
```

```
321     else m_CheckEnt4.SetCheck (0);
322     //Registro en la base de datos de los cambios
323     if (Ent1Aux!=Entrada1)
324     {
325         InsertarRegistro(Leer(&m_EditEntrada1),MirarEstado(9),InsertarHora());
326 
327     }
328     if (Ent2Aux!=Entrada2)
329     {
330         InsertarRegistro(Leer(&m_EditEntrada2),MirarEstado(10),InsertarHora());
331 
332     }
333     if (Ent3Aux!=Entrada3)
334     {
335         InsertarRegistro(Leer(&m_EditEntrada3),MirarEstado(11),InsertarHora());
336 
337     }
338     if (Ent4Aux!=Entrada4)
339     {
340         InsertarRegistro(Leer(&m_EditEntrada4),MirarEstado(12),InsertarHora());
341 
342     }
343     // Enviar Estado al socket si hay cambios
344     if((Ent1Aux!=Entrada1)||(Ent2Aux!=Entrada2)||(Ent3Aux!=Entrada3)||(Ent4Aux!=Entrada4))
345     {
346         if (m_Socket1.GetState ()==7) EnviarEstadoAlSck();
347     }
348 }
349 
350 void CProyectoBDDlg::EnviarEstadoAlSck()
351 {
352     CString Estado;
353 
354     Estado="Estado: ";
355 
356     if (Salida1==1) Estado.Insert (9,"1 ");
357     else Estado.Insert (9,"0 ");
358     if (Salida2==1) Estado.Insert (11,"1 ");
359     else Estado.Insert (11,"0 ");
360     if (Salida3==1) Estado.Insert (13,"1 ");
361     else Estado.Insert (13,"0 ");
362     if (Salida4==1) Estado.Insert (15,"1 ");
363     else Estado.Insert (15,"0 ");
364     if (Salida5==1) Estado.Insert (17,"1 ");
365     else Estado.Insert (17,"0 ");
366     if (Salida6==1) Estado.Insert (19,"1 ");
367     else Estado.Insert (19,"0 ");
368     if (Salida7==1) Estado.Insert (21,"1 ");
369     else Estado.Insert (21,"0 ");
370     if (Salida8==1) Estado.Insert (23,"1*");
371     else Estado.Insert (23,"0*");
372     if (Entrada1==1) Estado.Insert (25,"1 ");
373     else Estado.Insert (25,"0 ");
374     if (Entrada2==1) Estado.Insert (27,"1 ");
375     else Estado.Insert (27,"0 ");
376     if (Entrada3==1) Estado.Insert (29,"1 ");
377     else Estado.Insert (29,"0 ");
378     if (Entrada4==1) Estado.Insert (31,"1");
379     else Estado.Insert (31,"0");
380 
381     //m_Socket1.ABREnviarSck (Estado);
382     m_Socket1.ABREnviarSck (Estado);
383 
384 }
```

```
385
386 void CProyectoBDDlg::EnviarLeyendaAlSck()
387 {
388     CString Leyenda;
389     CString Estado;
390
391     Estado="Leyenda:";
392
393     Leyenda.Insert (Leyenda.GetLength (),Leer(&m_EditSalida1));
394     Leyenda.Insert (Leyenda.GetLength (),",");
395     Leyenda.Insert (Leyenda.GetLength (),Leer(&m_EditSalida2));
396     Leyenda.Insert (Leyenda.GetLength (),",");
397     Leyenda.Insert (Leyenda.GetLength (),Leer(&m_EditSalida3));
398     Leyenda.Insert (Leyenda.GetLength (),",");
399     Leyenda.Insert (Leyenda.GetLength (),Leer(&m_EditSalida4));
400     Leyenda.Insert (Leyenda.GetLength (),",");
401     Leyenda.Insert (Leyenda.GetLength (),Leer(&m_EditSalida5));
402     Leyenda.Insert (Leyenda.GetLength (),",");
403     Leyenda.Insert (Leyenda.GetLength (),Leer(&m_EditSalida6));
404     Leyenda.Insert (Leyenda.GetLength (),",");
405     Leyenda.Insert (Leyenda.GetLength (),Leer(&m_EditSalida7));
406     Leyenda.Insert (Leyenda.GetLength (),",");
407     Leyenda.Insert (Leyenda.GetLength (),Leer(&m_EditSalida8));
408     Leyenda.Insert (Leyenda.GetLength (),",");
409     Leyenda.Insert (Leyenda.GetLength (),Leer(&m_EditEntrada1));
410     Leyenda.Insert (Leyenda.GetLength (),",");
411     Leyenda.Insert (Leyenda.GetLength (),Leer(&m_EditEntrada2));
412     Leyenda.Insert (Leyenda.GetLength (),",");
413     Leyenda.Insert (Leyenda.GetLength (),Leer(&m_EditEntrada3));
414     Leyenda.Insert (Leyenda.GetLength (),",");
415     Leyenda.Insert (Leyenda.GetLength (),Leer(&m_EditEntrada4));
416     Leyenda.Insert (Leyenda.GetLength (),",");
417
418     if (Salida1==1) Estado.Insert (9,"1 ");
419     else Estado.Insert (9,"0 ");
420     if (Salida2==1) Estado.Insert (11,"1 ");
421     else Estado.Insert (11,"0 ");
422     if (Salida3==1) Estado.Insert (13,"1 ");
423     else Estado.Insert (13,"0 ");
424     if (Salida4==1) Estado.Insert (15,"1 ");
425     else Estado.Insert (15,"0 ");
426     if (Salida5==1) Estado.Insert (17,"1 ");
427     else Estado.Insert (17,"0 ");
428     if (Salida6==1) Estado.Insert (19,"1 ");
429     else Estado.Insert (19,"0 ");
430     if (Salida7==1) Estado.Insert (21,"1 ");
431     else Estado.Insert (21,"0 ");
432     if (Salida8==1) Estado.Insert (23,"1*");
433     else Estado.Insert (23,"0*");
434     if (Entrada1==1) Estado.Insert (25,"1 ");
435     else Estado.Insert (25,"0 ");
436     if (Entrada2==1) Estado.Insert (27,"1 ");
437     else Estado.Insert (27,"0 ");
438     if (Entrada3==1) Estado.Insert (29,"1 ");
439     else Estado.Insert (29,"0 ");
440     if (Entrada4==1) Estado.Insert (31,"1");
441     else Estado.Insert (31,"0");
442     Estado.Insert (Estado.GetLength (),"*");
443     Estado.Insert(Estado.GetLength (),Leyenda);
444
445     m_Socket1.ABREnviarSck (Estado);
446 }
447
448 void CProyectoBDDlg::ImprimirLeyenda()
```

```
449 {
450     CString aux;
451     m_EditSalida1.SetWindowText (AfxGetApp()->GetProfileString ("Leyenda","Salida1"));
452     m_EditSalida2.SetWindowText (AfxGetApp()->GetProfileString ("Leyenda","Salida2") );
453     m_EditSalida3.SetWindowText (AfxGetApp()->GetProfileString ("Leyenda","Salida3"));
454     m_EditSalida4.SetWindowText (AfxGetApp()->GetProfileString ("Leyenda","Salida4"));
455     m_EditSalida5.SetWindowText (AfxGetApp()->GetProfileString ("Leyenda","Salida5") );
456     m_EditSalida6.SetWindowText (AfxGetApp()->GetProfileString ("Leyenda","Salida6") );
457     m_EditSalida7.SetWindowText (AfxGetApp()->GetProfileString ("Leyenda","Salida7"));
458     m_EditSalida8.SetWindowText (AfxGetApp()->GetProfileString ("Leyenda","Salida8") );
459 
460     m_EditEntrada1.SetWindowText (AfxGetApp()->GetProfileString ("Leyenda","Entrada1"));
461     m_EditEntrada2.SetWindowText (AfxGetApp()->GetProfileString ("Leyenda","Entrada2") );
462     m_EditEntrada3.SetWindowText (AfxGetApp()->GetProfileString ("Leyenda","Entrada3") );
463     m_EditEntrada4.SetWindowText (AfxGetApp()->GetProfileString ("Leyenda","Entrada4") );
464 
465     m_EditIP.SetWindowText (m_Socket1.GetLocalIP () );
466     AfxGetApp()->WriteProfileString ("Socket","IP",m_Socket1.GetLocalIP ());
467 
468     m_EditPort.SetWindowText (AfxGetApp()->GetProfileString ("Socket","Port"));
469 
470     m_EditPath.SetWindowText (AfxGetApp()->GetProfileString ("BD","Path"));
471 
472     UpdateData(FALSE);
473 
474 }
```

```
1  void CBaseDatosSet::ABRAbrirBD()
2  {
3  //  m_pSet=&m_BaseDatosDlgSet;
4      if (IsOpen ()){
5          Close ();
6      }
7      Open ();
8
9  }
10
11 void CBaseDatosSet::ABRCerrarBD()
12 {
13     if (IsOpen ()){
14         Close ();
15     }
16 }
17
18 void CBaseDatosSet::ABRUltimoRegistro()
19 {
20     if ((IsOpen ())&&(!IsEOF())){
21         MoveLast();
22     }
23
24 }
25
26 void CBaseDatosSet::ABRPrimerRegistro()
27 {
28     if ((IsOpen ())&&(!IsBOF())){
29         MoveFirst();
30     }
31
32 }
33
34 void CBaseDatosSet::ABRSiguienteRegistro()
35 {
36
37     if ((IsOpen ())&&(!IsEOF())&&(IsBOF())){
38         MoveNext();
39     }
40     if ((IsOpen ())&&(!IsEOF())){
41         MoveNext();
42     }
43
44 }
45
46 void CBaseDatosSet::ABRAnteriorRegistro()
47 {
48     if ((IsOpen ())&&(!IsBOF())&&(IsEOF())){
49         MovePrev();
50     }
51     if ((IsOpen ())&&(!IsBOF())){
52         MovePrev();
53     }
54 }
55
56 void CBaseDatosSet::ABRInsertarRegistro(CString Nombre,CString Estado,CString Hora)
57 {
58     CString sqlcmd;
59
60 //  UpdateData(TRUE);
61
62     if (IsOpen ()){
63         sqlcmd.Format("INSERT INTO Historico (Nombre,Estado,Hora) VALUES ('%s','%s','%s')",Nomb
64         m_pDatabase->ExecuteSQL (sqlcmd);
```

```
65  //      ABRBuscarTodosLosRegistros();
66      }
67
68  }
69
70  void CBaseDatosSet::ABRBorrarRegistro(CString Nombre,CString Estado,CString Hora)
71  {
72      CString sqlcmd;
73
74  //  UpdateData(TRUE);
75
76      CString n="=";
77      CString e="=";
78      CString h="=";
79
80      if (Nombre=="") n="<>";
81      if (Estado=="") e="<>";
82      if (Hora=="") h="<>";
83
84
85
86      if (IsOpen ()){
87          sqlcmd.Format("DELETE FROM Historico WHERE (Nombre %s '%s' AND Estado %s '%s' AND Hora
88
89          if ((n==e) && (e==h) && (h=="<>"))
90          {
91              if (AfxMessageBox("¿Seguro que desea Borrar Todos Los Registros?", MB_YESNO)==IDYES
92              {
93                  m_pDatabase->ExecuteSQL (sqlcmd);
94              }
95
96          }
97          else m_pDatabase->ExecuteSQL (sqlcmd);
98
99  //      LimpiarCampos();
100
101 //      ABRBuscarTodosLosRegistros();
102     }
103
104 }
105
106 void CBaseDatosSet::ABRActualizarRegistro(CString Nombre,CString Estado,CString Hora)
107 {
108     CString sqlcmd;
109
110 //  UpdateData(TRUE);
111     if (IsOpen ()){
112         sqlcmd.Format("UPDATE Historico SET Nombre='%s',Estado='%s',Hora='%s' WHERE (Nombre='%s
113         m_pDatabase->ExecuteSQL (sqlcmd);
114 //      ABRBuscarTodosLosRegistros();
115     }
116
117 }
118
119 void CBaseDatosSet::ABRBuscarRegistro(CString Nombre,CString Estado,CString Hora)
120 {
121     CString sqlcmd,aux;
122
123 //  UpdateData(TRUE);
124
125     CString n="=";
126     CString e="=";
127     CString h="=";
128
```

```
129     if (Nombre=="") n="<>";
130     if (Estado=="") e="<>";
131     if (Hora=="") h="<>";
132 
133     if (IsOpen ()){
134 
135 //      sqlcmd.Format("SELECT COUNT (Hora) FROM Historico WHERE (Nombre %s '%s' AND Estado %s '
136 //      ABRCerrarBD();
137 //      Open (0,sqlcmd,NULL);
138 
139 //      aux.Format ("%s Registros encontrados.",m_pSet->m_Nombre);
140 //      m_Static1=aux;
141 //      UpdateData(FALSE);
142 
143         sqlcmd.Format("SELECT * FROM Historico WHERE (Nombre %s '%s' AND Estado %s '%s' AND Hor
144         //LimpiarCampos();
145         ABRCerrarBD();
146         Open (0,sqlcmd,NULL);
147 
148 //      UltimoRegistro();
149 //      LeerRegistro();
150     }
151 
152 }
153 
154 void CBaseDatosSet::ABRBuscarTodosLosRegistros()
155 {
156     CString sqlcmd;
157 
158     if (IsOpen ()){
159 
160         sqlcmd.Format("SELECT * FROM Historico");
161         ABRCerrarBD();
162         Open (0,sqlcmd,NULL);
163 
164         ABRUltimoRegistro();
165     }
166 
167 }
168 
169 
170 CString CBaseDatosSet::ABRContarRegistros()
171 {
172     CString sqlcmd;
173     CString aux;
174 
175     if (IsOpen ()){
176 
177         sqlcmd.Format("SELECT COUNT (Hora) FROM Historico");
178         ABRCerrarBD();
179         Open (0,sqlcmd,NULL);
180         aux.Format ("%s Registros en la Base de Datos.",m_Nombre);
181 
182         sqlcmd.Format("SELECT * FROM Historico");
183         ABRCerrarBD();
184         Open (0,sqlcmd,NULL);
185 
186     }
187 
188     return aux;
189 }
```

```
1  //Se deben incluir las declaraciones de las funciones de este fichero
2  //en el fichero MSWinsockControl.h
3
4  void CMSWinsockControl::ABRConectarSck()
5  {
6      // TODO: Add your control notification handler code here
7      CString m_host;
8      CString m_port;
9      VARIANT vtHost;
10     VARIANT vtPort;
11
12 //    if (AfxGetApp()->GetProfileString ("Socket2","IP")="")    AfxGetApp()->WriteProfileString
13 //  if (AfxGetApp()->GetProfileString ("Socket2","Port")="") AfxGetApp()->WriteProfileString ("
14
15     m_host=AfxGetApp()->GetProfileString ("Socket2","IP");
16     m_port=_T(AfxGetApp()->GetProfileString ("Socket2","Port"));
17
18     //m_host="127.0.0.1";
19     //m_port=_T("888");
20
21     vtHost.vt=VT_BSTR;
22     vtPort.vt=VT_BSTR;
23
24      SetRemoteHost(m_host);
25
26      SetRemotePort(atoi(m_port));
27
28     vtHost.bstrVal=m_host.AllocSysString();
29     vtPort.bstrVal=m_port.AllocSysString();
30
31      Connect(vtHost,vtPort);
32
33
34 }
35
36 void CMSWinsockControl::ABROnConnectWinsock1()
37 {
38     // TODO: Add your control notification handler code here
39
40 }
41
42 void CMSWinsockControl::ABREscucharSck()
43 {
44     // TODO: Add your command handler code here
45      Close ();
46      Listen ();
47
48
49 }
50
51 void CMSWinsockControl::ABROnConnectionRequestWinsock1(long requestID)
52 {
53     // TODO: Add your control notification handler code here
54      Close ();
55      Accept (requestID);
56
57
58 }
59
60 void CMSWinsockControl::ABROnDataArrivalWinsock1(long bytesTotal)
61 {
62     // TODO: Add your control notification handler code here
63     CString strport;
64     VARIANT vtData;
```

```
65      VARIANT vtType;
66      VARIANT vtMaxlen;
67
68      strport="4000";
69      vtData.vt=VT_BSTR;
70      vtType.vt=VT_ERROR;
71      vtMaxlen.vt=VT_I2;
72      vtMaxlen.iVal=(short)bytesTotal;
73
74
75      CString Datos;
76
77
78      vtData.bstrVal=strport.AllocSysString();
79      vtType.bstrVal=strport.AllocSysString();
80
81
82       GetData(&vtData,vtType,vtMaxlen);
83
84
85      Datos=vtData.bstrVal;
86
87      Dato=Datos;
88
89  }
90
91  void CMSWinsockControl::ABREnviarSck(CString datos)
92  {
93      // TODO: Add your control notification handler code here
94      VARIANT vtCommand;
95      CString m_command;
96
97      vtCommand.vt=VT_BSTR;
98      m_command=datos;
99
100     vtCommand.bstrVal =m_command.AllocSysString ();
101      SendData (vtCommand);
102
103
104 }
105
106 void CMSWinsockControl::ABROnCloseWinsock1()
107 {
108     // TODO: Add your control notification handler code here
109
110 }
111
112 void CMSWinsockControl::ABROnErrorWinsock1(short Number, BSTR FAR* Description, long Scode, LPC
113 {
114     // TODO: Add your control notification handler code here
115
116 }
117
118 void CMSWinsockControl::ABRCerrarSck()
119 {
120     // TODO: Add your command handler code here
121      Close ();
122
123 }
124
125 void CMSWinsockControl::ABROnSendCompleteWinsock1 ()
126 {
127     // TODO: Add your command handler code here
128
```

```
129 }
130
131
132 CString CMSWinsockControl::ABRConvertirEstado(int Estado)
133 {
134     switch(Estado)
135     {
136     case 0:
137         return "Closed";
138         break;
139     case 1 :
140         return "Open";
141         break;
142     case 2 :
143         return "Listening";
144         break;
145     case 3:
146         return "Connection pendingOpen";
147         break;
148     case 4 :
149         return "Resolving host";
150         break;
151     case 5 :
152         return "Host resolved";
153         break;
154     case 6 :
155         return "Connecting";
156         break;
157     case 7:
158         return "Connected";
159         break;
160     case 8 :
161         return "Peer is closing the connection";
162         break;
163     case 9 :
164         return "Error";
165         break;
166     default:
167         return "NULL";
168         break;
169     }
170
171 }
172
```

```cpp
// BaseDatosDlg.cpp : implementation file
//

#include "stdafx.h"
#include "ProyectoBD.h"
#include "BaseDatosDlg.h"

#include "ABRCBaseDatosDlgUtils.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

/////////////////////////////////////////////////////////////////////////////
// CBaseDatosDlg dialog

CBaseDatosDlg::CBaseDatosDlg(CWnd* pParent /*=NULL*/)
    : CDialog(CBaseDatosDlg::IDD, pParent)
{
    //{{AFX_DATA_INIT(CBaseDatosDlg)
    m_Static1 = _T("");
    //}}AFX_DATA_INIT
    Create(IDD);
}


void CBaseDatosDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CBaseDatosDlg)
    DDX_Control(pDX, IDC_EDIT_NOMBRE, m_EditNombre);
    DDX_Control(pDX, IDC_EDIT_HORA, m_EditHora);
    DDX_Control(pDX, IDC_EDIT_ESTADO, m_EditEstado);
    DDX_Text(pDX, IDC_STATIC1, m_Static1);
    //}}AFX_DATA_MAP
}


BEGIN_MESSAGE_MAP(CBaseDatosDlg, CDialog)
    //{{AFX_MSG_MAP(CBaseDatosDlg)
    ON_BN_CLICKED(IDC_BUTTON_PRIMERO, OnButtonPrimero)
    ON_BN_CLICKED(IDC_BUTTON_ANTERIOR, OnButtonAnterior)
    ON_BN_CLICKED(IDC_BUTTON_SIGUIENTE, OnButtonSiguiente)
    ON_BN_CLICKED(IDC_BUTTON_ULTIMO, OnButtonUltimo)
    ON_BN_CLICKED(IDC_BUTTON_INSERTAR, OnButtonInsertar)
    ON_BN_CLICKED(IDC_BUTTON_BORRAR, OnButtonBorrar)
    ON_BN_CLICKED(IDC_BUTTON_ACTUALIZAR, OnButtonActualizar)
    ON_BN_CLICKED(IDC_BUTTON_BUSCARTODOS, OnButtonBuscartodos)
    ON_BN_CLICKED(IDC_BUTTON_LIMPIAR, OnButtonLimpiar)
    ON_BN_CLICKED(IDC_BUTTON_BUSCAR, OnButtonBuscar)
    ON_WM_PAINT()
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()

/////////////////////////////////////////////////////////////////////////////
// CBaseDatosDlg message handlers


BOOL CBaseDatosDlg::OnInitDialog()
{
    CDialog::OnInitDialog();

```

```
65      // TODO: Add extra initialization here
66
67      m_pSet=&m_BaseDatosDlgSet ;
68      m_pSet->ABRCerrarBD ();
69      m_pSet->ABRAbrirBD ();
70  //  m_Static1=m_pSet->ABRContarRegistros();
71  //  UpdateData(FALSE);
72  //  m_pSet->ABRUltimoRegistro ();
73      OnButtonBuscartodos();
74      return TRUE;  // return TRUE unless you set the focus to a control
75                    // EXCEPTION: OCX Property Pages should return FALSE
76  }
77
78
79
80  void CBaseDatosDlg::OnPaint()
81  {
82      CPaintDC dc(this); // device context for painting
83
84      // TODO: Add your message handler code here
85
86  //  m_pSet=&m_BaseDatosDlgSet ;
87  //  m_pSet->ABRCerrarBD ();
88  //  m_pSet->ABRAbrirBD ();
89
90      EscribirRegistro();
91  //  OnButtonBuscartodos();
92      m_EditNombre.SetSel(m_EditNombre.GetWindowTextLength (),m_EditNombre.GetWindowTextLength ()
93      // Do not call CDialog::OnPaint() for painting messages
94  }
95
96  void CBaseDatosDlg::OnButtonPrimero()
97  {
98      // TODO: Add your control notification handler code here
99      m_pSet->ABRPrimerRegistro ();
100 //  BorrarCampos();
101     EscribirRegistro ();
102 }
103
104 void CBaseDatosDlg::OnButtonAnterior()
105 {
106     // TODO: Add your control notification handler code here
107     m_pSet->ABRAnteriorRegistro();
108 //  BorrarCampos();
109     EscribirRegistro ();
110 }
111
112 void CBaseDatosDlg::OnButtonSiguiente()
113 {
114     // TODO: Add your control notification handler code here
115     m_pSet->ABRSiguienteRegistro();
116 //  OnButtonLimpiar();
117     EscribirRegistro ();
118
119 }
120
121 void CBaseDatosDlg::OnButtonUltimo()
122 {
123     // TODO: Add your control notification handler code here
124
125     m_pSet->ABRUltimoRegistro();
126 //  OnButtonLimpiar();
127     EscribirRegistro ();
128
```

```
129 }
130
131 void CBaseDatosDlg::OnButtonInsertar()
132 {
133     // TODO: Add your control notification handler code here
134     CString aux;
135
136
137     if (Leer(&m_EditNombre)=="") m_pSet->m_Nombre="-";
138     if (Leer(&m_EditEstado)=="") m_pSet->m_Estado="-";
139     m_pSet->m_Hora=InsertarHora();
140     EscribirRegistro();
141
142     m_pSet->ABRInsertarRegistro(Leer(&m_EditNombre),Leer(&m_EditEstado),Leer(&m_EditHora));
143
144 //  m_Static1=m_pSet->ABRContarRegistros();
145 //  UpdateData(FALSE);
146
147     OnButtonBuscartodos();
148 }
149
150
151
152 void CBaseDatosDlg::OnButtonBorrar()
153 {
154     // TODO: Add your control notification handler code here
155     m_pSet->ABRBorrarRegistro (Leer(&m_EditNombre),Leer(&m_EditEstado),Leer(&m_EditHora));
156
157     OnButtonBuscartodos();
158
159     if (m_pSet->GetRecordCount ()==0)
160     {
161         m_pSet->ABRInsertarRegistro ("Inicio de la Base de Datos","-",InsertarHora());
162     }
163
164     OnButtonBuscartodos();
165     SetFocus();
166 }
167
168 void CBaseDatosDlg::OnButtonActualizar()
169 {
170     // TODO: Add your control notification handler code here
171     if((Leer(&m_EditNombre)!="")&&(Leer(&m_EditEstado)!="")&&(Leer(&m_EditHora)!=""))
172         m_pSet->ABRActualizarRegistro (Leer(&m_EditNombre),Leer(&m_EditEstado),Leer(&m_EditHora
173 }
174
175 void CBaseDatosDlg::OnButtonBuscartodos()
176 {
177     // TODO: Add your control notification handler code here
178     m_Static1=m_pSet->ABRContarRegistros();
179     UpdateData(FALSE);
180     m_pSet->m_Nombre ="";
181     m_pSet->m_Estado ="";
182     m_pSet->m_Hora ="";
183
184     m_pSet->ABRBuscarTodosLosRegistros ();
185 //  OnButtonUltimo();
186     EscribirRegistro ();
187 }
188
189 void CBaseDatosDlg::OnButtonLimpiar()
190 {
191     // TODO: Add your control notification handler code here
192     m_EditNombre.SetWindowText ("");
```

```
193     m_EditEstado.SetWindowText ("");
194     m_EditHora.SetWindowText ("");
195 //  m_pSet->m_Nombre="";
196 //  m_pSet->m_Estado="";
197 //  m_pSet->m_Hora="";
198 }
199
200 void CBaseDatosDlg::OnButtonBuscar()
201 {
202     // TODO: Add your control notification handler code here
203     m_pSet->m_Nombre="";
204     m_pSet->m_Estado="";
205     m_pSet->m_Hora="";
206
207     m_pSet->ABRBuscarRegistro (Leer(&m_EditNombre),Leer(&m_EditEstado),Leer(&m_EditHora));
208
209     OnButtonUltimo();
210
211 }
212
213
214
```

```
#if !defined(AFX_BASEDATOSDLG_H__9210F9FE_A9F7_11D5_97E1_FCCD2DB3034F__INCLUDED_)
#define AFX_BASEDATOSDLG_H__9210F9FE_A9F7_11D5_97E1_FCCD2DB3034F__INCLUDED_

#include "BaseDatosSet.h"   // Added by ClassView
#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000
// BaseDatosDlg.h : header file
//

/////////////////////////////////////////////////////////////////////////////
// CBaseDatosDlg dialog

class CBaseDatosSet;
class CBaseDatosDlg : public CDialog
{
// Construction
public:
    CString InsertarHora();
    void BorrarCampos();
    void EscribirRegistro ();
    CString Leer(CEdit *Edit);
    void Escribir(CEdit* Edit, CString String);

    CBaseDatosSet m_BaseDatosDlgSet;
    CBaseDatosSet* m_pSet;

    CBaseDatosDlg(CWnd* pParent = NULL);   // standard constructor

// Dialog Data
    //{{AFX_DATA(CBaseDatosDlg)
    enum { IDD = IDD_BASEDATOSDLG_DIALOG };
    CEdit   m_EditNombre;
    CEdit   m_EditHora;
    CEdit   m_EditEstado;
    CString m_Static1;
    //}}AFX_DATA


// Overrides
    // ClassWizard generated virtual function overrides
    //{{AFX_VIRTUAL(CBaseDatosDlg)
    public:
    protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV support
    //}}AFX_VIRTUAL

// Implementation
protected:

    // Generated message map functions
    //{{AFX_MSG(CBaseDatosDlg)
    afx_msg void OnButtonPrimero();
    virtual BOOL OnInitDialog();
    afx_msg void OnButtonAnterior();
    afx_msg void OnButtonSiguiente();
    afx_msg void OnButtonUltimo();
    afx_msg void OnButtonInsertar();
    afx_msg void OnButtonBorrar();
    afx_msg void OnButtonActualizar();
    afx_msg void OnButtonBuscartodos();
    afx_msg void OnButtonLimpiar();
    afx_msg void OnButtonBuscar();
    afx_msg void OnPaint();
```

```
65      //}}AFX_MSG
66      DECLARE_MESSAGE_MAP()
67  };
68
69  //{{AFX_INSERT_LOCATION}}
70  // Microsoft Visual C++ will insert additional declarations immediately before the previous line
71
72  #endif // !defined(AFX_BASEDATOSDLG_H__9210F9FE_A9F7_11D5_97E1_FCCD2DB3034F__INCLUDED_)
73
```

```cpp
// BaseDatosSet.cpp : implementation file
//

#include "stdafx.h"
#include "ProyectoBD.h"
#include "BaseDatosSet.h"

#include "ABRdb.h"


#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

/////////////////////////////////////////////////////////////////////////////
// CBaseDatosSet

IMPLEMENT_DYNAMIC(CBaseDatosSet, CRecordset)

CBaseDatosSet::CBaseDatosSet(CDatabase* pdb)
    : CRecordset(pdb)
{
    //{{AFX_FIELD_INIT(CBaseDatosSet)
    m_Nombre = _T("");
    m_Estado = _T("");
    m_Hora = _T("");
    m_nFields = 3;
    //}}AFX_FIELD_INIT
    m_nDefaultType = dynaset;
}


CString CBaseDatosSet::GetDefaultConnect()
{
//  return _T("ODBC;DRIVER={MICROSOFT ACCESS DRIVER (*.mdb)};DSN='';DBQ=C:\\windows\\escritorio\

    CString aux;
    if (AfxGetApp()->GetProfileString ("BD","Path")=="")
    {
        AfxGetApp()->WriteProfileString ("Socket","Port","888");
        AfxGetApp()->WriteProfileString ("BD","Path","bd1.mdb");

        AfxGetApp()->WriteProfileString ("Leyenda","Salida1","Salida1");
        AfxGetApp()->WriteProfileString ("Leyenda","Salida2","Salida2");
        AfxGetApp()->WriteProfileString ("Leyenda","Salida3","Salida3");
        AfxGetApp()->WriteProfileString ("Leyenda","Salida4","Salida4");
        AfxGetApp()->WriteProfileString ("Leyenda","Salida5","Salida5");
        AfxGetApp()->WriteProfileString ("Leyenda","Salida6","Salida6");
        AfxGetApp()->WriteProfileString ("Leyenda","Salida7","Salida7");
        AfxGetApp()->WriteProfileString ("Leyenda","Salida8","Salida8");

        AfxGetApp()->WriteProfileString ("Leyenda","Entrada1","Entrada1");
        AfxGetApp()->WriteProfileString ("Leyenda","Entrada2","Entrada2");
        AfxGetApp()->WriteProfileString ("Leyenda","Entrada3","Entrada3");
        AfxGetApp()->WriteProfileString ("Leyenda","Entrada4","Entrada4");

        AfxGetApp()->WriteProfileString ("Socket","IP","127.0.0.1");
    }
    aux="ODBC;DRIVER={MICROSOFT ACCESS DRIVER (*.mdb)};DSN='';DBQ=";
    aux.Insert (aux.GetLength (),AfxGetApp()->GetProfileString ("BD","Path"));
    return _T(aux);

```

```
65  //  return _T("ODBC;DRIVER={MICROSOFT ACCESS DRIVER (*.mdb)};DSN='';DBQ=C:\\windows\\escritorio\
66  //  En el path de la base de datos de arriba es necesario poner dos barras debido a estar entre
67
68  }
69
70  CString CBaseDatosSet::GetDefaultSQL()
71  {
72      return _T("[Historico]");
73  }
74
75  void CBaseDatosSet::DoFieldExchange(CFieldExchange* pFX)
76  {
77      //{{AFX_FIELD_MAP(CBaseDatosSet)
78      pFX->SetFieldType(CFieldExchange::outputColumn);
79      RFX_Text(pFX, _T("[Nombre]"), m_Nombre);
80      RFX_Text(pFX, _T("[Estado]"), m_Estado);
81      RFX_Text(pFX, _T("[Hora]"), m_Hora);
82      //}}AFX_FIELD_MAP
83  }
84
85  /////////////////////////////////////////////////////////////////////////////
86  // CBaseDatosSet diagnostics
87
88  #ifdef _DEBUG
89  void CBaseDatosSet::AssertValid() const
90  {
91      CRecordset::AssertValid();
92  }
93
94  void CBaseDatosSet::Dump(CDumpContext& dc) const
95  {
96      CRecordset::Dump(dc);
97  }
98  #endif //_DEBUG
99
```

```
#if !defined(AFX_BASEDATOSSET_H__9210F9FF_A9F7_11D5_97E1_FCCD2DB3034F__INCLUDED_)
#define AFX_BASEDATOSSET_H__9210F9FF_A9F7_11D5_97E1_FCCD2DB3034F__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000
// BaseDatosSet.h : header file
//

/////////////////////////////////////////////////////////////////////////////
// CBaseDatosSet recordset
class CBaseDatosSet : public CRecordset
{
public:
    CBaseDatosSet(CDatabase* pDatabase = NULL);
    DECLARE_DYNAMIC(CBaseDatosSet)

// Field/Param Data
    //{{AFX_FIELD(CBaseDatosSet, CRecordset)
    CString m_Nombre;
    CString m_Estado;
    CString m_Hora;
    //}}AFX_FIELD


// Overrides
    // ClassWizard generated virtual function overrides
    //{{AFX_VIRTUAL(CBaseDatosSet)
    public:
    virtual CString GetDefaultConnect();    // Default connection string
    virtual CString GetDefaultSQL();    // Default SQL for Recordset
    virtual void DoFieldExchange(CFieldExchange* pFX);  // RFX support

    //funciones Creadas por ABR
    void CBaseDatosSet::ABRAbrirBD();
    void CBaseDatosSet::ABRCerrarBD();
    void CBaseDatosSet::ABRUltimoRegistro();
    void CBaseDatosSet::ABRPrimerRegistro();
    void CBaseDatosSet::ABRSiguienteRegistro();
    void CBaseDatosSet::ABRAnteriorRegistro();
    void CBaseDatosSet::ABRInsertarRegistro(CString Nombre,CString Estado,CString Hora);
    void CBaseDatosSet::ABRBorrarRegistro(CString Nombre,CString Estado,CString Hora);
    void CBaseDatosSet::ABRActualizarRegistro(CString Nombre,CString Estado,CString Hora);
    void CBaseDatosSet::ABRBuscarRegistro(CString Nombre,CString Estado,CString Hora);
    void CBaseDatosSet::ABRBuscarTodosLosRegistros();
    CString CBaseDatosSet::ABRContarRegistros();

    // fin de las declaraciones
    //}}AFX_VIRTUAL

// Implementation
#ifdef _DEBUG
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
#endif
};

//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately before the previous line

#endif // !defined(AFX_BASEDATOSSET_H__9210F9FF_A9F7_11D5_97E1_FCCD2DB3034F__INCLUDED_)

```

```cpp
// ClienteDlg.cpp : implementation file
//

#include "stdafx.h"
#include "ProyectoBD.h"
#include "ClienteDlg.h"

#include "ABRCClienteDlgUtils.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

/////////////////////////////////////////////////////////////////////////////
// CClienteDlg dialog


CClienteDlg::CClienteDlg(CWnd* pParent /*=NULL*/)
    : CDialog(CClienteDlg::IDD, pParent)
{
    //{{AFX_DATA_INIT(CClienteDlg)
    m_Static1 = _T("");
    //}}AFX_DATA_INIT
}


void CClienteDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CClienteDlg)
    DDX_Control(pDX, IDC_CHK12, m_ChkEnt4);
    DDX_Control(pDX, IDC_CHK11, m_ChkEnt3);
    DDX_Control(pDX, IDC_CHK10, m_ChkEnt2);
    DDX_Control(pDX, IDC_CHK9, m_ChkEnt1);
    DDX_Control(pDX, IDC_CHK8, m_ChkSal8);
    DDX_Control(pDX, IDC_CHK6, m_ChkSal6);
    DDX_Control(pDX, IDC_CHK7, m_ChkSal7);
    DDX_Control(pDX, IDC_CHK5, m_ChkSal5);
    DDX_Control(pDX, IDC_CHK4, m_ChkSal4);
    DDX_Control(pDX, IDC_CHK3, m_ChkSal3);
    DDX_Control(pDX, IDC_CHK2, m_ChkSal2);
    DDX_Control(pDX, IDC_CHK1, m_ChkSal1);
    DDX_Control(pDX, IDC_BUTTON_CONECTAR, m_ButtonConectar);
    DDX_Control(pDX, IDC_BUTTON_CERRAR, m_ButtonCerrar);
    DDX_Control(pDX, IDC_EDIT17, m_EditEnviado);
    DDX_Control(pDX, IDC_EDIT16, m_EditRecibido);
    DDX_Control(pDX, IDC_EDIT014, m_EditPort);
    DDX_Control(pDX, IDC_EDIT013, m_EditIP);
    DDX_Control(pDX, IDC_EDIT012, m_EditEntrada4);
    DDX_Control(pDX, IDC_EDIT011, m_EditEntrada3);
    DDX_Control(pDX, IDC_EDIT010, m_EditEntrada2);
    DDX_Control(pDX, IDC_EDIT09, m_EditEntrada1);
    DDX_Control(pDX, IDC_EDIT08, m_EditSalida8);
    DDX_Control(pDX, IDC_EDIT07, m_EditSalida7);
    DDX_Control(pDX, IDC_EDIT06, m_EditSalida6);
    DDX_Control(pDX, IDC_EDIT05, m_EditSalida5);
    DDX_Control(pDX, IDC_EDIT04, m_EditSalida4);
    DDX_Control(pDX, IDC_EDIT03, m_EditSalida3);
    DDX_Control(pDX, IDC_EDIT02, m_EditSalida2);
    DDX_Control(pDX, IDC_EDIT01, m_EditSalida1);
    DDX_Control(pDX, IDC_WINSOCK1, m_Socket2);
    DDX_Text(pDX, IDC_STATIC1, m_Static1);
```

```
65      //}}AFX_DATA_MAP
66  }
67
68
69  BEGIN_MESSAGE_MAP(CClienteDlg, CDialog)
70      //{{AFX_MSG_MAP(CClienteDlg)
71      ON_BN_CLICKED(IDC_BUTTON_CONECTAR, OnButtonConectar)
72      ON_EN_CHANGE(IDC_EDIT013, OnChangeEdit013)
73      ON_EN_CHANGE(IDC_EDIT014, OnChangeEdit014)
74      ON_BN_CLICKED(IDC_BUTTON_CERRAR, OnButtonCerrar)
75      ON_BN_CLICKED(IDC_CHK1, OnChk1)
76      ON_BN_CLICKED(IDC_CHK2, OnChk2)
77      ON_BN_CLICKED(IDC_CHK3, OnChk3)
78      ON_BN_CLICKED(IDC_CHK4, OnChk4)
79      ON_BN_CLICKED(IDC_CHK5, OnChk5)
80      ON_BN_CLICKED(IDC_CHK6, OnChk6)
81      ON_BN_CLICKED(IDC_CHK7, OnChk7)
82      ON_BN_CLICKED(IDC_CHK8, OnChk8)
83      //}}AFX_MSG_MAP
84  END_MESSAGE_MAP()
85
86  /////////////////////////////////////////////////////////////////////////////
87  // CClienteDlg message handlers
88
89  BOOL CClienteDlg::OnInitDialog()
90  {
91      CDialog::OnInitDialog();
92
93      // TODO: Add extra initialization here
94      m_Static1=m_Socket2.ABRConvertirEstado (m_Socket2.GetState ());
95      UpdateData(FALSE);
96
97      if (AfxGetApp()->GetProfileString ("Socket2","IP")=="") AfxGetApp()->WriteProfileString ("S
98      if (AfxGetApp()->GetProfileString ("Socket2","Port")=="") AfxGetApp()->WriteProfileString (
99
100
101     m_EditIP.SetWindowText (AfxGetApp()->GetProfileString ("Socket2","IP"));
102     m_EditPort.SetWindowText (AfxGetApp()->GetProfileString ("Socket2","Port"));
103
104     m_ButtonCerrar.EnableWindow (FALSE);
105     m_ButtonConectar.EnableWindow (TRUE);
106
107     Salida1=0;
108     Salida2=0;
109     Salida3=0;
110     Salida4=0;
111     Salida5=0;
112     Salida6=0;
113     Salida7=0;
114     Salida8=0;
115
116     return TRUE;  // return TRUE unless you set the focus to a control
117                   // EXCEPTION: OCX Property Pages should return FALSE
118 }
119
120 void CClienteDlg::OnButtonConectar()
121 {
122     // TODO: Add your control notification handler code here
123     m_Socket2.ABRCerrarSck ();
124     m_Socket2.ABRConectarSck ();
125     m_Static1=m_Socket2.ABRConvertirEstado (m_Socket2.GetState ());
126     UpdateData(FALSE);
127     m_ButtonCerrar.EnableWindow (TRUE);
128     m_ButtonConectar.EnableWindow (FALSE);
```

```
129 }
130
131 void CClienteDlg::OnChangeEdit013()
132 {
133     // TODO: If this is a RICHEDIT control, the control will not
134     // send this notification unless you override the CDialog::OnInitDialog()
135     // function and call CRichEditCtrl().SetEventMask()
136     // with the ENM_CHANGE flag ORed into the mask.
137
138     // TODO: Add your control notification handler code here
139     AfxGetApp()->WriteProfileString ("Socket2","IP",Leer(&m_EditIP));
140 }
141
142 void CClienteDlg::OnChangeEdit014()
143 {
144     // TODO: If this is a RICHEDIT control, the control will not
145     // send this notification unless you override the CDialog::OnInitDialog()
146     // function and call CRichEditCtrl().SetEventMask()
147     // with the ENM_CHANGE flag ORed into the mask.
148
149     // TODO: Add your control notification handler code here
150     AfxGetApp()->WriteProfileString ("Socket2","Port",Leer(&m_EditPort));
151
152 }
153
154
155 BEGIN_EVENTSINK_MAP(CClienteDlg, CDialog)
156     //{{AFX_EVENTSINK_MAP(CClienteDlg)
157     ON_EVENT(CClienteDlg, IDC_WINSOCK1, 1 /* Connect */, OnConnectWinsock1, VTS_NONE)
158     ON_EVENT(CClienteDlg, IDC_WINSOCK1, 0 /* DataArrival */, OnDataArrivalWinsock1, VTS_I4)
159     ON_EVENT(CClienteDlg, IDC_WINSOCK1, 5 /* Close */, OnCloseWinsock1, VTS_NONE)
160     //}}AFX_EVENTSINK_MAP
161 END_EVENTSINK_MAP()
162
163 void CClienteDlg::OnConnectWinsock1()
164 {
165     // TODO: Add your control notification handler code here
166     m_Static1=m_Socket2.ABRConvertirEstado (m_Socket2.GetState ());
167     UpdateData(FALSE);
168
169 }
170
171 void CClienteDlg::OnButtonCerrar()
172 {
173     // TODO: Add your control notification handler code here
174     m_Socket2.ABRCerrarSck ();
175     m_Static1=m_Socket2.ABRConvertirEstado (m_Socket2.GetState ());
176     UpdateData(FALSE);
177     m_ButtonCerrar.EnableWindow (FALSE);
178     m_ButtonConectar.EnableWindow (TRUE);
179 }
180
181 void CClienteDlg::OnDataArrivalWinsock1(long bytesTotal)
182 {
183     // TODO: Add your control notification handler code here
184     m_Socket2.ABROnDataArrivalWinsock1 (bytesTotal);
185
186
187     CString aux;
188     aux=m_Socket2.Dato;
189     aux=aux.Mid (0,7);
190
191     if (aux=="Estado:") InicializarSalidas();
192     if (aux=="Leyenda") InicializarLeyenda();
```

```
193     else Escribir(&m_EditRecibido,m_Socket2.Dato );
194
195 /*  if (m_Socket1.Dato=="Salida1") OnSalidasSalida1();
196     if (m_Socket1.Dato=="Salida2") OnSalidasSalida2();
197     if (m_Socket1.Dato=="Salida3") OnSalidasSalida3();
198     if (m_Socket1.Dato=="Salida4") OnSalidasSalida4();
199     if (m_Socket1.Dato=="Salida5") OnSalidasSalida5();
200     if (m_Socket1.Dato=="Salida6") OnSalidasSalida6();
201     if (m_Socket1.Dato=="Salida7") OnSalidasSalida7();
202     if (m_Socket1.Dato=="Salida8") OnSalidasSalida8();*/
203 }
204
205
206 void CClienteDlg::OnChk1()
207 {
208     // TODO: Add your control notification handler code here
209     if (m_Socket2.GetState ()==7)
210     {
211         Salida1=!Salida1;
212         m_Socket2.ABREnviarSck ("Salida1");
213         Escribir(&m_EditEnviado,Leer(&m_EditSalida1));
214     }
215     else m_ChkSal1.SetCheck (0);
216 }
217
218 void CClienteDlg::OnChk2()
219 {
220     // TODO: Add your control notification handler code here
221     if (m_Socket2.GetState ()==7)
222     {
223         Salida2=!Salida2;
224         m_Socket2.ABREnviarSck ("Salida2");
225         Escribir(&m_EditEnviado,Leer(&m_EditSalida2));
226     }
227     else m_ChkSal2.SetCheck (0);
228 }
229
230 void CClienteDlg::OnChk3()
231 {
232     // TODO: Add your control notification handler code here
233     if (m_Socket2.GetState ()==7)
234     {
235         Salida3=!Salida3;
236         m_Socket2.ABREnviarSck ("Salida3");
237         Escribir(&m_EditEnviado,Leer(&m_EditSalida3));
238     }
239     else m_ChkSal3.SetCheck (0);
240 }
241
242 void CClienteDlg::OnChk4()
243 {
244     // TODO: Add your control notification handler code here
245     if (m_Socket2.GetState ()==7)
246     {
247         Salida4=!Salida4;
248         m_Socket2.ABREnviarSck ("Salida4");
249         Escribir(&m_EditEnviado,Leer(&m_EditSalida4));
250     }
251     else m_ChkSal4.SetCheck (0);
252 }
253
254 void CClienteDlg::OnChk5()
255 {
256     // TODO: Add your control notification handler code here
```

```
257     if (m_Socket2.GetState ()==7)
258     {
259         Salida5=!Salida5;
260         m_Socket2.ABREnviarSck ("Salida5");
261         Escribir(&m_EditEnviado,Leer(&m_EditSalida5));
262     }
263     else m_ChkSal5.SetCheck (0);
264 }
265 
266 void CClienteDlg::OnChk6()
267 {
268     // TODO: Add your control notification handler code here
269     if (m_Socket2.GetState ()==7)
270     {
271         Salida6=!Salida6;
272         m_Socket2.ABREnviarSck ("Salida6");
273         Escribir(&m_EditEnviado,Leer(&m_EditSalida6));
274     }
275     else m_ChkSal6.SetCheck (0);
276 }
277 
278 void CClienteDlg::OnChk7()
279 {
280     // TODO: Add your control notification handler code here
281     if (m_Socket2.GetState ()==7)
282     {
283         Salida7=!Salida7;
284         m_Socket2.ABREnviarSck ("Salida7");
285         Escribir(&m_EditEnviado,Leer(&m_EditSalida7));
286     }
287     else m_ChkSal7.SetCheck (0);
288 }
289 
290 void CClienteDlg::OnChk8()
291 {
292     // TODO: Add your control notification handler code here
293     if (m_Socket2.GetState ()==7)
294     {
295         Salida8=!Salida8;
296         m_Socket2.ABREnviarSck ("Salida8");
297         Escribir(&m_EditEnviado,Leer(&m_EditSalida8));
298     }
299     else m_ChkSal8.SetCheck (0);
300 }
301 
302 void CClienteDlg::OnCloseWinsock1()
303 {
304     // TODO: Add your control notification handler code here
305     m_Socket2.ABRCerrarSck ();
306     m_Static1=m_Socket2.ABRConvertirEstado (m_Socket2.GetState ());
307     UpdateData(FALSE);
308     m_ButtonCerrar.EnableWindow (FALSE);
309     m_ButtonConectar.EnableWindow (TRUE);
310 }
311 
```

```
1  //{{AFX_INCLUDES()
2  #include "mswinsockcontrol.h"
3  //}}AFX_INCLUDES
4  #if !defined(AFX_CLIENTEDLG_H__D341D442_B0DF_11D5_97E1_B4A0DF43684E__INCLUDED_)
5  #define AFX_CLIENTEDLG_H__D341D442_B0DF_11D5_97E1_B4A0DF43684E__INCLUDED_
6
7  #if _MSC_VER > 1000
8  #pragma once
9  #endif // _MSC_VER > 1000
10 // ClienteDlg.h : header file
11 //
12
13 /////////////////////////////////////////////////////////////////////////////
14 // CClienteDlg dialog
15
16 class CClienteDlg : public CDialog
17 {
18 // Construction
19 public:
20     void InicializarLeyenda();
21     void ActualizarPuerto();
22     void InicializarSalidas();
23     void Escribir(CEdit* Edit, CString String);
24     CString Leer(CEdit* Edit);
25     int Salida1;
26     int Salida2;
27     int Salida3;
28     int Salida4;
29     int Salida5;
30     int Salida6;
31     int Salida7;
32     int Salida8;
33     int Entrada1;
34     int Entrada2;
35     int Entrada3;
36     int Entrada4;
37
38     CClienteDlg(CWnd* pParent = NULL);   // standard constructor
39
40 // Dialog Data
41     //{{AFX_DATA(CClienteDlg)
42     enum { IDD = IDD_CLIENTEDLG_DIALOG };
43     CButton m_ChkEnt4;
44     CButton m_ChkEnt3;
45     CButton m_ChkEnt2;
46     CButton m_ChkEnt1;
47     CButton m_ChkSal8;
48     CButton m_ChkSal6;
49     CButton m_ChkSal7;
50     CButton m_ChkSal5;
51     CButton m_ChkSal4;
52     CButton m_ChkSal3;
53     CButton m_ChkSal2;
54     CButton m_ChkSal1;
55     CButton m_ButtonConectar;
56     CButton m_ButtonCerrar;
57     CEdit   m_EditEnviado;
58     CEdit   m_EditRecibido;
59     CEdit   m_EditPort;
60     CEdit   m_EditIP;
61     CEdit   m_EditEntrada4;
62     CEdit   m_EditEntrada3;
63     CEdit   m_EditEntrada2;
64     CEdit   m_EditEntrada1;
```

```
65      CEdit   m_EditSalida8;
66      CEdit   m_EditSalida7;
67      CEdit   m_EditSalida6;
68      CEdit   m_EditSalida5;
69      CEdit   m_EditSalida4;
70      CEdit   m_EditSalida3;
71      CEdit   m_EditSalida2;
72      CEdit   m_EditSalida1;
73      CMSWinsockControl   m_Socket2;
74      CString m_Static1;
75      //}}AFX_DATA
76
77
78  // Overrides
79      // ClassWizard generated virtual function overrides
80      //{{AFX_VIRTUAL(CClienteDlg)
81      protected:
82      virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV support
83      //}}AFX_VIRTUAL
84
85  // Implementation
86  protected:
87
88      // Generated message map functions
89      //{{AFX_MSG(CClienteDlg)
90      virtual BOOL OnInitDialog();
91      afx_msg void OnButtonConectar();
92      afx_msg void OnChangeEdit013();
93      afx_msg void OnChangeEdit014();
94      afx_msg void OnConnectWinsock1();
95      afx_msg void OnButtonCerrar();
96      afx_msg void OnDataArrivalWinsock1(long bytesTotal);
97      afx_msg void OnChk1();
98      afx_msg void OnChk2();
99      afx_msg void OnChk3();
100     afx_msg void OnChk4();
101     afx_msg void OnChk5();
102     afx_msg void OnChk6();
103     afx_msg void OnChk7();
104     afx_msg void OnChk8();
105     afx_msg void OnCloseWinsock1();
106     DECLARE_EVENTSINK_MAP()
107     //}}AFX_MSG
108     DECLARE_MESSAGE_MAP()
109 };
110
111 //{{AFX_INSERT_LOCATION}}
112 // Microsoft Visual C++ will insert additional declarations immediately before the previous lin
113
114 #endif // !defined(AFX_CLIENTEDLG_H__D341D442_B0DF_11D5_97E1_B4A0DF43684E__INCLUDED_)
115
```

```cpp
// Machine generated IDispatch wrapper class(es) created by Microsoft Visual C++

// NOTE: Do not modify the contents of this file.  If this class is regenerated by
//  Microsoft Visual C++, your modifications will be overwritten.


#include "stdafx.h"
#include "mswinsockcontrol.h"

#include "ABRWinsock.h"

/////////////////////////////////////////////////////////////////////////////
// CMSWinsockControl

IMPLEMENT_DYNCREATE(CMSWinsockControl, CWnd)

/////////////////////////////////////////////////////////////////////////////
// CMSWinsockControl properties

/////////////////////////////////////////////////////////////////////////////
// CMSWinsockControl operations

long CMSWinsockControl::GetProtocol()
{
    long result;
    InvokeHelper(0x3, DISPATCH_PROPERTYGET, VT_I4, (void*)&result, NULL);
    return result;
}

void CMSWinsockControl::SetProtocol(long nNewValue)
{
    static BYTE parms[] =
        VTS_I4;
    InvokeHelper(0x3, DISPATCH_PROPERTYPUT, VT_EMPTY, NULL, parms,
         nNewValue);
}

CString CMSWinsockControl::GetRemoteHostIP()
{
    CString result;
    InvokeHelper(0x4, DISPATCH_PROPERTYGET, VT_BSTR, (void*)&result, NULL);
    return result;
}

CString CMSWinsockControl::GetLocalHostName()
{
    CString result;
    InvokeHelper(0x5, DISPATCH_PROPERTYGET, VT_BSTR, (void*)&result, NULL);
    return result;
}

CString CMSWinsockControl::GetLocalIP()
{
    CString result;
    InvokeHelper(0x6, DISPATCH_PROPERTYGET, VT_BSTR, (void*)&result, NULL);
    return result;
}

long CMSWinsockControl::GetSocketHandle()
{
    long result;
    InvokeHelper(0x7, DISPATCH_PROPERTYGET, VT_I4, (void*)&result, NULL);
    return result;
}
```

```
 65
 66 long CMSWinsockControl::GetRemotePort()
 67 {
 68     long result;
 69     InvokeHelper(0x1, DISPATCH_PROPERTYGET, VT_I4, (void*)&result, NULL);
 70     return result;
 71 }
 72
 73 void CMSWinsockControl::SetRemotePort(long nNewValue)
 74 {
 75     static BYTE parms[] =
 76         VTS_I4;
 77     InvokeHelper(0x1, DISPATCH_PROPERTYPUT, VT_EMPTY, NULL, parms,
 78          nNewValue);
 79 }
 80
 81 long CMSWinsockControl::GetLocalPort()
 82 {
 83     long result;
 84     InvokeHelper(0x2, DISPATCH_PROPERTYGET, VT_I4, (void*)&result, NULL);
 85     return result;
 86 }
 87
 88 void CMSWinsockControl::SetLocalPort(long nNewValue)
 89 {
 90     static BYTE parms[] =
 91         VTS_I4;
 92     InvokeHelper(0x2, DISPATCH_PROPERTYPUT, VT_EMPTY, NULL, parms,
 93          nNewValue);
 94 }
 95
 96 short CMSWinsockControl::GetState()
 97 {
 98     short result;
 99     InvokeHelper(0x8, DISPATCH_PROPERTYGET, VT_I2, (void*)&result, NULL);
100     return result;
101 }
102
103 long CMSWinsockControl::GetBytesReceived()
104 {
105     long result;
106     InvokeHelper(0x9, DISPATCH_PROPERTYGET, VT_I4, (void*)&result, NULL);
107     return result;
108 }
109
110 void CMSWinsockControl::Connect(const VARIANT& RemoteHost, const VARIANT& RemotePort)
111 {
112     static BYTE parms[] =
113         VTS_VARIANT VTS_VARIANT;
114     InvokeHelper(0x40, DISPATCH_METHOD, VT_EMPTY, NULL, parms,
115          &RemoteHost, &RemotePort);
116 }
117
118 void CMSWinsockControl::Listen()
119 {
120     InvokeHelper(0x41, DISPATCH_METHOD, VT_EMPTY, NULL, NULL);
121 }
122
123 void CMSWinsockControl::Accept(long requestID)
124 {
125     static BYTE parms[] =
126         VTS_I4;
127     InvokeHelper(0x42, DISPATCH_METHOD, VT_EMPTY, NULL, parms,
128          requestID);
```

```
129 }
130
131 void CMSWinsockControl::SendData(const VARIANT& data)
132 {
133     static BYTE parms[] =
134         VTS_VARIANT;
135     InvokeHelper(0x43, DISPATCH_METHOD, VT_EMPTY, NULL, parms,
136          &data);
137 }
138
139 void CMSWinsockControl::GetData(VARIANT* data, const VARIANT& type, const VARIANT& maxLen)
140 {
141     static BYTE parms[] =
142         VTS_PVARIANT VTS_VARIANT VTS_VARIANT;
143     InvokeHelper(0x44, DISPATCH_METHOD, VT_EMPTY, NULL, parms,
144          data, &type, &maxLen);
145 }
146
147 void CMSWinsockControl::PeekData(VARIANT* data, const VARIANT& type, const VARIANT& maxLen)
148 {
149     static BYTE parms[] =
150         VTS_PVARIANT VTS_VARIANT VTS_VARIANT;
151     InvokeHelper(0x45, DISPATCH_METHOD, VT_EMPTY, NULL, parms,
152          data, &type, &maxLen);
153 }
154
155 void CMSWinsockControl::Close()
156 {
157     InvokeHelper(0x46, DISPATCH_METHOD, VT_EMPTY, NULL, NULL);
158 }
159
160 void CMSWinsockControl::Bind(const VARIANT& LocalPort, const VARIANT& LocalIP)
161 {
162     static BYTE parms[] =
163         VTS_VARIANT VTS_VARIANT;
164     InvokeHelper(0x47, DISPATCH_METHOD, VT_EMPTY, NULL, parms,
165          &LocalPort, &LocalIP);
166 }
167
168 CString CMSWinsockControl::GetRemoteHost()
169 {
170     CString result;
171     InvokeHelper(0xa, DISPATCH_PROPERTYGET, VT_BSTR, (void*)&result, NULL);
172     return result;
173 }
174
175 void CMSWinsockControl::SetRemoteHost(LPCTSTR lpszNewValue)
176 {
177     static BYTE parms[] =
178         VTS_BSTR;
179     InvokeHelper(0xa, DISPATCH_PROPERTYPUT, VT_EMPTY, NULL, parms,
180          lpszNewValue);
181 }
182
183
184
```

```
#if !defined(AFX_MSWINSOCKCONTROL_H__5F886C4F_A2C9_11D5_97E1_CFB3C086B260__INCLUDED_)
#define AFX_MSWINSOCKCONTROL_H__5F886C4F_A2C9_11D5_97E1_CFB3C086B260__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000
// Machine generated IDispatch wrapper class(es) created by Microsoft Visual C++

// NOTE: Do not modify the contents of this file.  If this class is regenerated by
//  Microsoft Visual C++, your modifications will be overwritten.

/////////////////////////////////////////////////////////////////////////////
// CMSWinsockControl wrapper class

class CMSWinsockControl : public CWnd
{
protected:
    DECLARE_DYNCREATE(CMSWinsockControl)
public:
    CLSID const& GetClsid()
    {
        static CLSID const clsid
            = { 0x248dd896, 0xbb45, 0x11cf, { 0x9a, 0xbc, 0x0, 0x80, 0xc7, 0xe7, 0xb7, 0x8d } };
        return clsid;
    }
    virtual BOOL Create(LPCTSTR lpszClassName,
        LPCTSTR lpszWindowName, DWORD dwStyle,
        const RECT& rect,
        CWnd* pParentWnd, UINT nID,
        CCreateContext* pContext = NULL)
    { return CreateControl(GetClsid(), lpszWindowName, dwStyle, rect, pParentWnd, nID); }

    BOOL Create(LPCTSTR lpszWindowName, DWORD dwStyle,
        const RECT& rect, CWnd* pParentWnd, UINT nID,
        CFile* pPersist = NULL, BOOL bStorage = FALSE,
        BSTR bstrLicKey = NULL)
    { return CreateControl(GetClsid(), lpszWindowName, dwStyle, rect, pParentWnd, nID,
        pPersist, bStorage, bstrLicKey); }

// Attributes
public:

// Operations
public:
     //Variable para recoger los datos recibidos con el DataArrival
    CString Dato;

    //Declaraciones de las funciones del fichero ABRWinsock.h

    void CMSWinsockControl::ABRConectarSck();
    void CMSWinsockControl::ABROnConnectWinsock1();
    void CMSWinsockControl::ABREscucharSck();
    void CMSWinsockControl::ABROnConnectionRequestWinsock1(long requestID);
    void CMSWinsockControl::ABROnDataArrivalWinsock1(long bytesTotal);
    void CMSWinsockControl::ABREnviarSck(CString Datos);
    void CMSWinsockControl::ABROnCloseWinsock1();
    void CMSWinsockControl::ABROnErrorWinsock1(short Number, BSTR FAR* Description, long Scode,
    void CMSWinsockControl::ABRCerrarSck();
    void CMSWinsockControl::ABROnSendCompleteWinsock1();
    CString CMSWinsockControl::ABRConvertirEstado(int Estado);
    //Fin de las declaraciones

    long GetProtocol();
    void SetProtocol(long nNewValue);
```

```
65      CString GetRemoteHostIP();
66      CString GetLocalHostName();
67      CString GetLocalIP();
68      long GetSocketHandle();
69      long GetRemotePort();
70      void SetRemotePort(long nNewValue);
71      long GetLocalPort();
72      void SetLocalPort(long nNewValue);
73      short GetState();
74      long GetBytesReceived();
75      void Connect(const VARIANT& RemoteHost, const VARIANT& RemotePort);
76      void Listen();
77      void Accept(long requestID);
78      void SendData(const VARIANT& data);
79      void GetData(VARIANT* data, const VARIANT& type, const VARIANT& maxLen);
80      void PeekData(VARIANT* data, const VARIANT& type, const VARIANT& maxLen);
81      void Close();
82      void Bind(const VARIANT& LocalPort, const VARIANT& LocalIP);
83      CString GetRemoteHost();
84      void SetRemoteHost(LPCTSTR lpszNewValue);
85  };
86
87  //{{AFX_INSERT_LOCATION}}
88  // Microsoft Visual C++ will insert additional declarations immediately before the previous line
89
90  #endif // !defined(AFX_MSWINSOCKCONTROL_H__5F886C4F_A2C9_11D5_97E1_CFB3C086B260__INCLUDED_)
91
```

```cpp
// Propiedades1Dlg.cpp : implementation file
//

#include "stdafx.h"
#include "ProyectoBD.h"
#include "Propiedades1Dlg.h"


#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

/////////////////////////////////////////////////////////////////////////////
// CPropiedades1Dlg dialog


CPropiedades1Dlg::CPropiedades1Dlg(CWnd* pParent /*=NULL*/)
    : CDialog(CPropiedades1Dlg::IDD, pParent)
{
    //{{AFX_DATA_INIT(CPropiedades1Dlg)
    m_EditSalida1 = _T("");
    m_EditSalida2 = _T("");
    m_EditSalida3 = _T("");
    m_EditSalida4 = _T("");
    m_EditSalida5 = _T("");
    m_EditSalida6 = _T("");
    m_EditSalida7 = _T("");
    m_EditSalida8 = _T("");
    m_EditEntrada1 = _T("");
    m_EditEntrada2 = _T("");
    m_EditEntrada3 = _T("");
    m_EditEntrada4 = _T("");
    m_EditIP = _T("");
    m_EditPort = _T("");
    m_EditPath = _T("");
    //}}AFX_DATA_INIT
}


void CPropiedades1Dlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CPropiedades1Dlg)
    DDX_Text(pDX, IDC_EDIT1, m_EditSalida1);
    DDX_Text(pDX, IDC_EDIT2, m_EditSalida2);
    DDX_Text(pDX, IDC_EDIT3, m_EditSalida3);
    DDX_Text(pDX, IDC_EDIT4, m_EditSalida4);
    DDX_Text(pDX, IDC_EDIT5, m_EditSalida5);
    DDX_Text(pDX, IDC_EDIT6, m_EditSalida6);
    DDX_Text(pDX, IDC_EDIT7, m_EditSalida7);
    DDX_Text(pDX, IDC_EDIT8, m_EditSalida8);
    DDX_Text(pDX, IDC_EDIT9, m_EditEntrada1);
    DDX_Text(pDX, IDC_EDIT10, m_EditEntrada2);
    DDX_Text(pDX, IDC_EDIT11, m_EditEntrada3);
    DDX_Text(pDX, IDC_EDIT12, m_EditEntrada4);
    DDX_Text(pDX, IDC_EDIT13, m_EditIP);
    DDX_Text(pDX, IDC_EDIT14, m_EditPort);
    DDX_Text(pDX, IDC_EDIT15, m_EditPath);
    //}}AFX_DATA_MAP
}


```

```
65  BEGIN_MESSAGE_MAP(CPropiedades1Dlg, CDialog)
66      //{{AFX_MSG_MAP(CPropiedades1Dlg)
67      //}}AFX_MSG_MAP
68  END_MESSAGE_MAP()
69
70  /////////////////////////////////////////////////////////////////////////////
71  // CPropiedades1Dlg message handlers
72
73  void CPropiedades1Dlg::OnOK()
74  {
75      // TODO: Add extra validation here
76      UpdateData(TRUE);
77
78      AfxGetApp()->WriteProfileString ("Leyenda","Salida1",m_EditSalida1);
79      AfxGetApp()->WriteProfileString ("Leyenda","Salida2",m_EditSalida2);
80      AfxGetApp()->WriteProfileString ("Leyenda","Salida3",m_EditSalida3);
81      AfxGetApp()->WriteProfileString ("Leyenda","Salida4",m_EditSalida4);
82      AfxGetApp()->WriteProfileString ("Leyenda","Salida5",m_EditSalida5);
83      AfxGetApp()->WriteProfileString ("Leyenda","Salida6",m_EditSalida6);
84      AfxGetApp()->WriteProfileString ("Leyenda","Salida7",m_EditSalida7);
85      AfxGetApp()->WriteProfileString ("Leyenda","Salida8",m_EditSalida8);
86
87      AfxGetApp()->WriteProfileString ("Leyenda","Entrada1",m_EditEntrada1);
88      AfxGetApp()->WriteProfileString ("Leyenda","Entrada2",m_EditEntrada2);
89      AfxGetApp()->WriteProfileString ("Leyenda","Entrada3",m_EditEntrada3);
90      AfxGetApp()->WriteProfileString ("Leyenda","Entrada4",m_EditEntrada4);
91
92      AfxGetApp()->WriteProfileString ("Socket","IP",m_EditIP);
93      AfxGetApp()->WriteProfileString ("Socket","Port",m_EditPort);
94
95      AfxGetApp()->WriteProfileString ("BD","Path",m_EditPath);
96
97      CDialog::OnOK();
98  }
99
100 BOOL CPropiedades1Dlg::OnInitDialog()
101 {
102     CDialog::OnInitDialog();
103
104     // TODO: Add extra initialization here
105
106     m_EditSalida1=AfxGetApp()->GetProfileString ("Leyenda","Salida1");
107     m_EditSalida2=AfxGetApp()->GetProfileString ("Leyenda","Salida2");
108     m_EditSalida3=AfxGetApp()->GetProfileString ("Leyenda","Salida3");
109     m_EditSalida4=AfxGetApp()->GetProfileString ("Leyenda","Salida4");
110     m_EditSalida5=AfxGetApp()->GetProfileString ("Leyenda","Salida5");
111     m_EditSalida6=AfxGetApp()->GetProfileString ("Leyenda","Salida6");
112     m_EditSalida7=AfxGetApp()->GetProfileString ("Leyenda","Salida7");
113     m_EditSalida8=AfxGetApp()->GetProfileString ("Leyenda","Salida8");
114
115     m_EditEntrada1=AfxGetApp()->GetProfileString ("Leyenda","Entrada1");
116     m_EditEntrada2=AfxGetApp()->GetProfileString ("Leyenda","Entrada2");
117     m_EditEntrada3=AfxGetApp()->GetProfileString ("Leyenda","Entrada3");
118     m_EditEntrada4=AfxGetApp()->GetProfileString ("Leyenda","Entrada4");
119
120     m_EditIP=AfxGetApp()->GetProfileString ("Socket","IP");
121     m_EditPort=AfxGetApp()->GetProfileString ("Socket","Port");
122
123     m_EditPath=AfxGetApp()->GetProfileString ("BD","Path");
124
125     UpdateData(FALSE);
126
127     return TRUE;  // return TRUE unless you set the focus to a control
128                   // EXCEPTION: OCX Property Pages should return FALSE
```

```
129 }
130
```

```
#if !defined(AFX_PROPIEDADES1DLG_H__6159E6C1_AF94_11D5_97E1_EE6A57BB0840__INCLUDED_)
#define AFX_PROPIEDADES1DLG_H__6159E6C1_AF94_11D5_97E1_EE6A57BB0840__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000
// Propiedades1Dlg.h : header file
//

/////////////////////////////////////////////////////////////////////////////
// CPropiedades1Dlg dialog

class CPropiedades1Dlg : public CDialog
{
// Construction
public:
    CPropiedades1Dlg(CWnd* pParent = NULL);   // standard constructor

// Dialog Data
    //{{AFX_DATA(CPropiedades1Dlg)
    enum { IDD = IDD_PROPIEDADES_DIALOG };
    CString m_EditSalida1;
    CString m_EditSalida2;
    CString m_EditSalida3;
    CString m_EditSalida4;
    CString m_EditSalida5;
    CString m_EditSalida6;
    CString m_EditSalida7;
    CString m_EditSalida8;
    CString m_EditEntrada1;
    CString m_EditEntrada2;
    CString m_EditEntrada3;
    CString m_EditEntrada4;
    CString m_EditIP;
    CString m_EditPort;
    CString m_EditPath;
    //}}AFX_DATA


// Overrides
    // ClassWizard generated virtual function overrides
    //{{AFX_VIRTUAL(CPropiedades1Dlg)
    protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV support
    //}}AFX_VIRTUAL

// Implementation
protected:

    // Generated message map functions
    //{{AFX_MSG(CPropiedades1Dlg)
    virtual void OnOK();
    virtual BOOL OnInitDialog();
    //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};

//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately before the previous line

#endif // !defined(AFX_PROPIEDADES1DLG_H__6159E6C1_AF94_11D5_97E1_EE6A57BB0840__INCLUDED_)

```

```
; CLW file contains information for the MFC ClassWizard

[General Info]
Version=1
LastClass=CProyectoBDDlg
LastTemplate=CDialog
NewFileInclude1=#include "stdafx.h"
NewFileInclude2=#include "ProyectoBD.h"

ClassCount=7
Class1=CProyectoBDApp
Class2=CProyectoBDDlg
Class3=CAboutDlg

ResourceCount=7
Resource1=IDD_BASEDATOSDLG_DIALOG
Resource2=IDR_MAINFRAME
Resource3=IDD_PROPIEDADES_DIALOG
Resource4=IDD_ABOUTBOX
Class4=CBaseDatosDlg
Class5=CBaseDatosSet
Resource5=IDD_CLIENTEDLG_DIALOG
Class6=CPropiedades1Dlg
Resource6=IDD_PROYECTOBD_DIALOG
Class7=CClienteDlg
Resource7=IDR_MENU1

[CLS:CProyectoBDApp]
Type=0
HeaderFile=ProyectoBD.h
ImplementationFile=ProyectoBD.cpp
Filter=N

[CLS:CProyectoBDDlg]
Type=0
HeaderFile=ProyectoBDDlg.h
ImplementationFile=ProyectoBDDlg.cpp
Filter=D
BaseClass=CDialog
VirtualFilter=dWC
LastObject=ID_ACCIN_BASEDEDATOS

[CLS:CAboutDlg]
Type=0
HeaderFile=ProyectoBDDlg.h
ImplementationFile=ProyectoBDDlg.cpp
Filter=D

[DLG:IDD_ABOUTBOX]
Type=1
Class=CAboutDlg
ControlCount=4
Control1=IDC_STATIC,static,1342177283
Control2=IDC_STATIC,static,1342308480
Control3=IDC_STATIC,static,1342308352
Control4=IDOK,button,1342373889

[DLG:IDD_PROYECTOBD_DIALOG]
Type=1
Class=CProyectoBDDlg
ControlCount=56
Control1=IDC_STATIC1,static,1342308352
Control2=IDC_EDIT_NOMBRE,edit,1350631552
Control3=IDC_EDIT_ESTADO,edit,1350631552
```

```
Control4=IDC_EDIT_HORA,edit,1350631552
Control5=IDC_STATIC,static,1342308354
Control6=IDC_STATIC,static,1342308354
Control7=IDC_STATIC,static,1342308354
Control8=IDC_CHECK1,button,1342242883
Control9=IDC_CHECK2,button,1342242883
Control10=IDC_CHECK3,button,1342242883
Control11=IDC_CHECK4,button,1342242883
Control12=IDC_CHECK5,button,1342242883
Control13=IDC_CHECK6,button,1342242883
Control14=IDC_CHECK7,button,1342242883
Control15=IDC_CHECK8,button,1342242883
Control16=IDC_STATIC,button,1342177287
Control17=IDC_WINSOCK1,{248DD896-BB45-11CF-9ABC-0080C7E7B78D},1342242816
Control18=IDC_CHECK9,button,1476460611
Control19=IDC_CHECK10,button,1476460611
Control20=IDC_CHECK11,button,1476460611
Control21=IDC_CHECK12,button,1476460611
Control22=IDC_STATIC,button,1342177287
Control23=IDC_STATIC,static,1342308352
Control24=IDC_EDIT01,edit,1350633600
Control25=IDC_STATIC,static,1342308352
Control26=IDC_EDIT02,edit,1350633600
Control27=IDC_STATIC,static,1342308352
Control28=IDC_EDIT03,edit,1350633600
Control29=IDC_STATIC,static,1342308352
Control30=IDC_EDIT04,edit,1350633600
Control31=IDC_STATIC,static,1342308352
Control32=IDC_EDIT05,edit,1350633600
Control33=IDC_STATIC,static,1342308352
Control34=IDC_EDIT06,edit,1350633600
Control35=IDC_STATIC,static,1342308352
Control36=IDC_EDIT07,edit,1350633600
Control37=IDC_STATIC,static,1342308352
Control38=IDC_EDIT08,edit,1350633600
Control39=IDC_STATIC,static,1342308352
Control40=IDC_EDIT09,edit,1350633600
Control41=IDC_STATIC,static,1342308352
Control42=IDC_EDIT010,edit,1350633600
Control43=IDC_STATIC,static,1342308352
Control44=IDC_EDIT011,edit,1350633600
Control45=IDC_STATIC,static,1342308352
Control46=IDC_EDIT012,edit,1350633600
Control47=IDC_STATIC,static,1342308352
Control48=IDC_EDIT013,edit,1350633600
Control49=IDC_STATIC,static,1342308352
Control50=IDC_EDIT014,edit,1350633600
Control51=IDC_STATIC,button,1342177287
Control52=IDC_STATIC,button,1342177287
Control53=IDC_STATIC,button,1342177287
Control54=IDC_STATIC,static,1342308352
Control55=IDC_EDIT16,edit,1350633600
Control56=IDC_STATIC,button,1342177287

[DLG:IDD_BASEDATOSDLG_DIALOG]
Type=1
Class=CBaseDatosDlg
ControlCount=17
Control1=IDC_STATIC1,static,1342308352
Control2=IDC_EDIT_NOMBRE,edit,1350631552
Control3=IDC_EDIT_ESTADO,edit,1350631552
Control4=IDC_EDIT_HORA,edit,1350631552
Control5=IDC_BUTTON_PRIMERO,button,1342242816
Control6=IDC_BUTTON_ANTERIOR,button,1342242816
```

```
Control7=IDC_BUTTON_SIGUIENTE,button,1342242816
Control8=IDC_BUTTON_ULTIMO,button,1342242816
Control9=IDC_BUTTON_INSERTAR,button,1342242816
Control10=IDC_BUTTON_BORRAR,button,1342242816
Control11=IDC_BUTTON_ACTUALIZAR,button,1342242816
Control12=IDC_BUTTON_BUSCARTODOS,button,1342242816
Control13=IDC_BUTTON_LIMPIAR,button,1342242816
Control14=IDC_BUTTON_BUSCAR,button,1342242816
Control15=IDC_STATIC,static,1342308354
Control16=IDC_STATIC,static,1342308354
Control17=IDC_STATIC,static,1342308354

[CLS:CBaseDatosDlg]
Type=0
HeaderFile=BaseDatosDlg.h
ImplementationFile=BaseDatosDlg.cpp
BaseClass=CDialog
Filter=D
VirtualFilter=dWC
LastObject=CBaseDatosDlg

[MNU:IDR_MENU1]
Type=1
Class=?
Command1=ID_ACCIN_BASEDEDATOS
Command2=ID_ACCIN_PROPIEDADES
Command3=ID_ACCIN_CLIENTE
Command4=ID_SALIDAS_SALIDA1
Command5=ID_SALIDAS_SALIDA2
Command6=ID_SALIDAS_SALIDA3
Command7=ID_SALIDAS_SALIDA4
Command8=ID_SALIDAS_SALIDA5
Command9=ID_SALIDAS_SALIDA6
Command10=ID_SALIDAS_SALIDA7
Command11=ID_SALIDAS_SALIDA8
CommandCount=11

[CLS:CBaseDatosSet]
Type=0
HeaderFile=BaseDatosSet.h
ImplementationFile=BaseDatosSet.cpp
BaseClass=CRecordset
Filter=N
VirtualFilter=r

[DB:CBaseDatosSet]
DB=1
DBType=ODBC
ColumnCount=3
Column1=[Nombre], 1, 64
Column2=[Estado], 1, 64
Column3=[Hora], 1, 64

[DLG:IDD_PROPIEDADES_DIALOG]
Type=1
Class=CPropiedades1Dlg
ControlCount=36
Control1=IDOK,button,1342242817
Control2=IDCANCEL,button,1342242816
Control3=IDC_STATIC,static,1342308352
Control4=IDC_EDIT1,edit,1350631552
Control5=IDC_STATIC,static,1342308352
Control6=IDC_EDIT2,edit,1350631552
Control7=IDC_STATIC,static,1342308352
```

```
Control8=IDC_EDIT3,edit,1350631552
Control9=IDC_STATIC,static,1342308352
Control10=IDC_EDIT4,edit,1350631552
Control11=IDC_STATIC,static,1342308352
Control12=IDC_EDIT5,edit,1350631552
Control13=IDC_STATIC,static,1342308352
Control14=IDC_EDIT6,edit,1350631552
Control15=IDC_STATIC,static,1342308352
Control16=IDC_EDIT7,edit,1350631552
Control17=IDC_STATIC,static,1342308352
Control18=IDC_EDIT8,edit,1350631552
Control19=IDC_STATIC,static,1342308352
Control20=IDC_EDIT9,edit,1350631552
Control21=IDC_STATIC,static,1342308352
Control22=IDC_EDIT10,edit,1350631552
Control23=IDC_STATIC,static,1342308352
Control24=IDC_EDIT11,edit,1350631552
Control25=IDC_STATIC,static,1342308352
Control26=IDC_EDIT12,edit,1350631552
Control27=IDC_STATIC,static,1342308352
Control28=IDC_EDIT13,edit,1350631552
Control29=IDC_STATIC,static,1342308352
Control30=IDC_EDIT14,edit,1350631552
Control31=IDC_STATIC,button,1342177287
Control32=IDC_STATIC,button,1342177287
Control33=IDC_STATIC,button,1342177287
Control34=IDC_EDIT15,edit,1350631552
Control35=IDC_STATIC,static,1342308352
Control36=IDC_STATIC,button,1342177287

[CLS:CPropiedades1Dlg]
Type=0
HeaderFile=Propiedades1Dlg.h
ImplementationFile=Propiedades1Dlg.cpp
BaseClass=CDialog
Filter=D
LastObject=CPropiedades1Dlg
VirtualFilter=dWC

[DLG:IDD_CLIENTEDLG_DIALOG]
Type=1
Class=CClienteDlg
ControlCount=54
Control1=IDC_STATIC1,static,1342308352
Control2=IDC_STATIC,button,1342177287
Control3=IDC_WINSOCK1,{248DD896-BB45-11CF-9ABC-0080C7E7B78D},1342242816
Control4=IDC_STATIC,button,1342177287
Control5=IDC_STATIC,static,1342308352
Control6=IDC_EDIT01,edit,1350633600
Control7=IDC_STATIC,static,1342308352
Control8=IDC_EDIT02,edit,1350633600
Control9=IDC_STATIC,static,1342308352
Control10=IDC_EDIT03,edit,1350633600
Control11=IDC_STATIC,static,1342308352
Control12=IDC_EDIT04,edit,1350633600
Control13=IDC_STATIC,static,1342308352
Control14=IDC_EDIT05,edit,1350633600
Control15=IDC_STATIC,static,1342308352
Control16=IDC_EDIT06,edit,1350633600
Control17=IDC_STATIC,static,1342308352
Control18=IDC_EDIT07,edit,1350633600
Control19=IDC_STATIC,static,1342308352
Control20=IDC_EDIT08,edit,1350633600
Control21=IDC_STATIC,static,1342308352
```

```
Control22=IDC_EDIT09,edit,1350633600
Control23=IDC_STATIC,static,1342308352
Control24=IDC_EDIT010,edit,1350633600
Control25=IDC_STATIC,static,1342308352
Control26=IDC_EDIT011,edit,1350633600
Control27=IDC_STATIC,static,1342308352
Control28=IDC_EDIT012,edit,1350633600
Control29=IDC_STATIC,static,1342308352
Control30=IDC_EDIT013,edit,1350631552
Control31=IDC_STATIC,static,1342308352
Control32=IDC_EDIT014,edit,1350631552
Control33=IDC_STATIC,button,1342177287
Control34=IDC_STATIC,button,1342177287
Control35=IDC_STATIC,static,1342308352
Control36=IDC_EDIT16,edit,1352730756
Control37=IDC_STATIC,button,1342177287
Control38=IDC_BUTTON_CONECTAR,button,1342242816
Control39=IDC_STATIC,static,1342308352
Control40=IDC_EDIT17,edit,1352730756
Control41=IDC_BUTTON_CERRAR,button,1342242816
Control42=IDC_STATIC,button,1342177287
Control43=IDC_CHK1,button,1342242883
Control44=IDC_CHK2,button,1342242883
Control45=IDC_CHK3,button,1342242883
Control46=IDC_CHK4,button,1342242883
Control47=IDC_CHK5,button,1342242883
Control48=IDC_CHK6,button,1342242883
Control49=IDC_CHK7,button,1342242883
Control50=IDC_CHK8,button,1342242883
Control51=IDC_CHK9,button,1476460611
Control52=IDC_CHK10,button,1476460611
Control53=IDC_CHK11,button,1476460611
Control54=IDC_CHK12,button,1476460611

[CLS:CClienteDlg]
Type=0
HeaderFile=ClienteDlg.h
ImplementationFile=ClienteDlg.cpp
BaseClass=CDialog
Filter=D
VirtualFilter=dWC
LastObject=IDC_WINSOCK1
```

```cpp
// ProyectoBD.cpp : Defines the class behaviors for the application.
//

#include "stdafx.h"
#include "ProyectoBD.h"
#include "ProyectoBDDlg.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

/////////////////////////////////////////////////////////////////////////////
// CProyectoBDApp

BEGIN_MESSAGE_MAP(CProyectoBDApp, CWinApp)
    //{{AFX_MSG_MAP(CProyectoBDApp)
        // NOTE - the ClassWizard will add and remove mapping macros here.
        //    DO NOT EDIT what you see in these blocks of generated code!
    //}}AFX_MSG
    ON_COMMAND(ID_HELP, CWinApp::OnHelp)
END_MESSAGE_MAP()

/////////////////////////////////////////////////////////////////////////////
// CProyectoBDApp construction

CProyectoBDApp::CProyectoBDApp()
{
    // TODO: add construction code here,
    // Place all significant initialization in InitInstance
}

/////////////////////////////////////////////////////////////////////////////
// The one and only CProyectoBDApp object

CProyectoBDApp theApp;

/////////////////////////////////////////////////////////////////////////////
// CProyectoBDApp initialization

BOOL CProyectoBDApp::InitInstance()
{
    AfxEnableControlContainer();

    // Standard initialization
    // If you are not using these features and wish to reduce the size
    //  of your final executable, you should remove from the following
    //  the specific initialization routines you do not need.

#ifdef _AFXDLL
    Enable3dControls();         // Call this when using MFC in a shared DLL
#else
    Enable3dControlsStatic();   // Call this when linking to MFC statically
#endif
    SetRegistryKey("ABRProyectos"); //Entradas al Registro ABR

    CProyectoBDDlg dlg;
    m_pMainWnd = &dlg;
    int nResponse = dlg.DoModal();
    if (nResponse == IDOK)
    {
        // TODO: Place code here to handle when the dialog is
        //  dismissed with OK
```

```
65      }
66      else if (nResponse == IDCANCEL)
67      {
68          // TODO: Place code here to handle when the dialog is
69          //  dismissed with Cancel
70      }
71
72      // Since the dialog has been closed, return FALSE so that we exit the
73      //  application, rather than start the application's message pump.
74      return FALSE;
75  }
76
```

```cpp
// ProyectoBDDlg.cpp : implementation file
//

#include "stdafx.h"
#include "ProyectoBD.h"
#include "ProyectoBDDlg.h"

#include "BaseDatosDlg.h"
#include "ABRCProyectoBDDlgUtils.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

/////////////////////////////////////////////////////////////////////////////
// CAboutDlg dialog used for App About

class CAboutDlg : public CDialog
{
public:
    CAboutDlg();

// Dialog Data
    //{{AFX_DATA(CAboutDlg)
    enum { IDD = IDD_ABOUTBOX };
    //}}AFX_DATA

    // ClassWizard generated virtual function overrides
    //{{AFX_VIRTUAL(CAboutDlg)
    protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV support
    //}}AFX_VIRTUAL

// Implementation
protected:
    //{{AFX_MSG(CAboutDlg)
    //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)
{
    //{{AFX_DATA_INIT(CAboutDlg)
    //}}AFX_DATA_INIT
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CAboutDlg)
    //}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
    //{{AFX_MSG_MAP(CAboutDlg)
        // No message handlers
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()

/////////////////////////////////////////////////////////////////////////////
// CProyectoBDDlg dialog

```

```
65  CProyectoBDDlg::CProyectoBDDlg(CWnd* pParent /*=NULL*/)
66      : CDialog(CProyectoBDDlg::IDD, pParent)
67  {
68      //{{AFX_DATA_INIT(CProyectoBDDlg)
69      m_Static1 = _T("");
70      //}}AFX_DATA_INIT
71      // Note that LoadIcon does not require a subsequent DestroyIcon in Win32
72      m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
73      m_pBaseDatosDlg = new CBaseDatosDlg();
74  }
75
76  void CProyectoBDDlg::DoDataExchange(CDataExchange* pDX)
77  {
78      CDialog::DoDataExchange(pDX);
79      //{{AFX_DATA_MAP(CProyectoBDDlg)
80      DDX_Control(pDX, IDC_EDIT16, m_EditPath);
81      DDX_Control(pDX, IDC_EDIT014, m_EditPort);
82      DDX_Control(pDX, IDC_EDIT013, m_EditIP);
83      DDX_Control(pDX, IDC_EDIT012, m_EditEntrada4);
84      DDX_Control(pDX, IDC_EDIT011, m_EditEntrada3);
85      DDX_Control(pDX, IDC_EDIT010, m_EditEntrada2);
86      DDX_Control(pDX, IDC_EDIT09, m_EditEntrada1);
87      DDX_Control(pDX, IDC_EDIT08, m_EditSalida8);
88      DDX_Control(pDX, IDC_EDIT07, m_EditSalida7);
89      DDX_Control(pDX, IDC_EDIT06, m_EditSalida6);
90      DDX_Control(pDX, IDC_EDIT05, m_EditSalida5);
91      DDX_Control(pDX, IDC_EDIT04, m_EditSalida4);
92      DDX_Control(pDX, IDC_EDIT03, m_EditSalida3);
93      DDX_Control(pDX, IDC_EDIT02, m_EditSalida2);
94      DDX_Control(pDX, IDC_EDIT01, m_EditSalida1);
95      DDX_Control(pDX, IDC_CHECK12, m_CheckEnt4);
96      DDX_Control(pDX, IDC_CHECK10, m_CheckEnt2);
97      DDX_Control(pDX, IDC_CHECK11, m_CheckEnt3);
98      DDX_Control(pDX, IDC_CHECK9, m_CheckEnt1);
99      DDX_Control(pDX, IDC_CHECK8, m_CheckSal8);
100     DDX_Control(pDX, IDC_CHECK7, m_CheckSal7);
101     DDX_Control(pDX, IDC_CHECK6, m_CheckSal6);
102     DDX_Control(pDX, IDC_CHECK5, m_CheckSal5);
103     DDX_Control(pDX, IDC_CHECK4, m_CheckSal4);
104     DDX_Control(pDX, IDC_CHECK3, m_CheckSal3);
105     DDX_Control(pDX, IDC_CHECK2, m_CheckSal2);
106     DDX_Control(pDX, IDC_CHECK1, m_CheckSal1);
107     DDX_Control(pDX, IDC_EDIT_NOMBRE, m_EditNombre);
108     DDX_Control(pDX, IDC_EDIT_HORA, m_EditHora);
109     DDX_Control(pDX, IDC_EDIT_ESTADO, m_EditEstado);
110     DDX_Text(pDX, IDC_STATIC1, m_Static1);
111     DDX_Control(pDX, IDC_WINSOCK1, m_Socket1);
112     //}}AFX_DATA_MAP
113 }
114
115 BEGIN_MESSAGE_MAP(CProyectoBDDlg, CDialog)
116     //{{AFX_MSG_MAP(CProyectoBDDlg)
117     ON_WM_SYSCOMMAND()
118     ON_WM_PAINT()
119     ON_WM_QUERYDRAGICON()
120     ON_COMMAND(ID_ACCIN_BASEDEDATOS, OnAccinBasededatos)
121     ON_COMMAND(ID_SALIDAS_SALIDA1, OnSalidasSalida1)
122     ON_COMMAND(ID_SALIDAS_SALIDA2, OnSalidasSalida2)
123     ON_COMMAND(ID_SALIDAS_SALIDA3, OnSalidasSalida3)
124     ON_COMMAND(ID_SALIDAS_SALIDA4, OnSalidasSalida4)
125     ON_COMMAND(ID_SALIDAS_SALIDA5, OnSalidasSalida5)
126     ON_COMMAND(ID_SALIDAS_SALIDA6, OnSalidasSalida6)
127     ON_COMMAND(ID_SALIDAS_SALIDA7, OnSalidasSalida7)
128     ON_COMMAND(ID_SALIDAS_SALIDA8, OnSalidasSalida8)
```

```
129     ON_WM_DESTROY()
130     ON_BN_CLICKED(IDC_CHECK1, OnCheck1)
131     ON_BN_CLICKED(IDC_CHECK2, OnCheck2)
132     ON_BN_CLICKED(IDC_CHECK3, OnCheck3)
133     ON_BN_CLICKED(IDC_CHECK4, OnCheck4)
134     ON_BN_CLICKED(IDC_CHECK5, OnCheck5)
135     ON_BN_CLICKED(IDC_CHECK6, OnCheck6)
136     ON_BN_CLICKED(IDC_CHECK7, OnCheck7)
137     ON_BN_CLICKED(IDC_CHECK8, OnCheck8)
138     ON_WM_TIMER()
139     ON_COMMAND(ID_ACCIN_PROPIEDADES, OnAccinPropiedades)
140     ON_COMMAND(ID_ACCIN_CLIENTE, OnAccinCliente)
141     //}}AFX_MSG_MAP
142 END_MESSAGE_MAP()
143 
144 /////////////////////////////////////////////////////////////////////////////
145 // CProyectoBDDlg message handlers
146 
147 BOOL CProyectoBDDlg::OnInitDialog()
148 {
149     CDialog::OnInitDialog();
150 
151     // Add "About..." menu item to system menu.
152 
153     // IDM_ABOUTBOX must be in the system command range.
154     ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
155     ASSERT(IDM_ABOUTBOX < 0xF000);
156 
157     CMenu* pSysMenu = GetSystemMenu(FALSE);
158     if (pSysMenu != NULL)
159     {
160         CString strAboutMenu;
161         strAboutMenu.LoadString(IDS_ABOUTBOX);
162         if (!strAboutMenu.IsEmpty())
163         {
164             pSysMenu->AppendMenu(MF_SEPARATOR);
165             pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
166         }
167     }
168 
169     // Set the icon for this dialog.  The framework does this automatically
170     //  when the application's main window is not a dialog
171     SetIcon(m_hIcon, TRUE);         // Set big icon
172     SetIcon(m_hIcon, FALSE);        // Set small icon
173 
174     // TODO: Add extra initialization here
175 
176 
177     m_nTimer1=SetTimer(1,500,NULL);
178 
179     m_pSet=&m_BaseDatosDlgSet;
180 
181     m_pSet->ABRAbrirBD ();
182 
183     InicializarSalidas();
184     m_pSet->ABRCerrarBD ();
185 
186 
187     ImprimirLeyenda();
188 
189 
190 
191     m_Socket1.SetLocalPort (atoi(Leer(&m_EditPort)));
192     m_Socket1.ABREscucharSck ();
```

```
193     m_Static1=m_Socket1.ABRConvertirEstado (m_Socket1.GetState ());
194     UpdateData(FALSE);
195
196     return TRUE;  // return TRUE  unless you set the focus to a control
197 }
198
199 void CProyectoBDDlg::OnSysCommand(UINT nID, LPARAM lParam)
200 {
201     if ((nID & 0xFFF0) == IDM_ABOUTBOX)
202     {
203         CAboutDlg dlgAbout;
204         dlgAbout.DoModal();
205     }
206     else
207     {
208         CDialog::OnSysCommand(nID, lParam);
209     }
210 }
211
212 // If you add a minimize button to your dialog, you will need the code below
213 //  to draw the icon.  For MFC applications using the document/view model,
214 //  this is automatically done for you by the framework.
215
216 void CProyectoBDDlg::OnPaint()
217 {
218     if (IsIconic())
219     {
220         CPaintDC dc(this); // device context for painting
221
222         SendMessage(WM_ICONERASEBKGND, (WPARAM) dc.GetSafeHdc(), 0);
223
224         // Center icon in client rectangle
225         int cxIcon = GetSystemMetrics(SM_CXICON);
226         int cyIcon = GetSystemMetrics(SM_CYICON);
227         CRect rect;
228         GetClientRect(&rect);
229         int x = (rect.Width() - cxIcon + 1) / 2;
230         int y = (rect.Height() - cyIcon + 1) / 2;
231
232         // Draw the icon
233         dc.DrawIcon(x, y, m_hIcon);
234     }
235     else
236     {
237         CDialog::OnPaint();
238     }
239     m_EditNombre.SetSel(m_EditNombre.GetWindowTextLength (),m_EditNombre.GetWindowTextLength ()
240
241
242 }
243
244 // The system calls this to obtain the cursor to display while the user drags
245 //  the minimized window.
246 HCURSOR CProyectoBDDlg::OnQueryDragIcon()
247 {
248     return (HCURSOR) m_hIcon;
249 }
250
251 void CProyectoBDDlg::OnDestroy()
252 {
253     CDialog::OnDestroy();
254
255     // TODO: Add your message handler code here
256     m_pSet=&m_BaseDatosDlgSet;
```

```
257     m_pSet->ABRAbrirBD ();
258
259     GuardarSalidas();
260
261     m_pSet->ABRCerrarBD ();
262
263 }
264
265 void CProyectoBDDlg::OnAccinBasededatos()
266 {
267     // TODO: Add your command handler code here
268 //  BaseDatosDlg.DoModal ();
269     m_pBaseDatosDlg->ShowWindow(SW_SHOW);
270     m_pBaseDatosDlg->SetFocus ();
271 }
272
273 void CProyectoBDDlg::OnSalidasSalida1()
274 {
275     // TODO: Add your command handler code here
276     InsertarRegistro(Leer(&m_EditSalida1),MirarEstado(1),InsertarHora());
277     ActualizarPuerto();
278 }
279
280 void CProyectoBDDlg::OnSalidasSalida2()
281 {
282     // TODO: Add your command handler code here
283         InsertarRegistro(Leer(&m_EditSalida2),MirarEstado(2),InsertarHora());
284         ActualizarPuerto();
285 }
286
287 void CProyectoBDDlg::OnSalidasSalida3()
288 {
289     // TODO: Add your command handler code here
290         InsertarRegistro(Leer(&m_EditSalida3),MirarEstado(3),InsertarHora());
291         ActualizarPuerto();
292 }
293
294 void CProyectoBDDlg::OnSalidasSalida4()
295 {
296     // TODO: Add your command handler code here
297         InsertarRegistro(Leer(&m_EditSalida4),MirarEstado(4),InsertarHora());
298         ActualizarPuerto();
299 }
300
301 void CProyectoBDDlg::OnSalidasSalida5()
302 {
303     // TODO: Add your command handler code here
304         InsertarRegistro(Leer(&m_EditSalida5),MirarEstado(5),InsertarHora());
305         ActualizarPuerto();
306 }
307
308 void CProyectoBDDlg::OnSalidasSalida6()
309 {
310     // TODO: Add your command handler code here
311         InsertarRegistro(Leer(&m_EditSalida6),MirarEstado(6),InsertarHora());
312         ActualizarPuerto();
313 }
314
315 void CProyectoBDDlg::OnSalidasSalida7()
316 {
317     // TODO: Add your command handler code here
318         InsertarRegistro(Leer(&m_EditSalida7),MirarEstado(7),InsertarHora());
319         ActualizarPuerto();
320 }
```

```
321
322 void CProyectoBDDlg::OnSalidasSalida8()
323 {
324     // TODO: Add your command handler code here
325         InsertarRegistro(Leer(&m_EditSalida8),MirarEstado(8),InsertarHora());
326         ActualizarPuerto();
327 }
328
329
330
331
332 void CProyectoBDDlg::OnCheck1()
333 {
334     // TODO: Add your control notification handler code here
335     OnSalidasSalida1();
336 }
337
338 void CProyectoBDDlg::OnCheck2()
339 {
340     // TODO: Add your control notification handler code here
341     OnSalidasSalida2();
342 }
343
344 void CProyectoBDDlg::OnCheck3()
345 {
346     // TODO: Add your control notification handler code here
347     OnSalidasSalida3();
348 }
349
350 void CProyectoBDDlg::OnCheck4()
351 {
352     // TODO: Add your control notification handler code here
353     OnSalidasSalida4();
354 }
355
356 void CProyectoBDDlg::OnCheck5()
357 {
358     // TODO: Add your control notification handler code here
359     OnSalidasSalida5();
360 }
361
362 void CProyectoBDDlg::OnCheck6()
363 {
364     // TODO: Add your control notification handler code here
365     OnSalidasSalida6();
366 }
367
368 void CProyectoBDDlg::OnCheck7()
369 {
370     // TODO: Add your control notification handler code here
371     OnSalidasSalida7();
372 }
373
374 void CProyectoBDDlg::OnCheck8()
375 {
376     // TODO: Add your control notification handler code here
377     OnSalidasSalida8();
378 }
379
380 BEGIN_EVENTSINK_MAP(CProyectoBDDlg, CDialog)
381     //{{AFX_EVENTSINK_MAP(CProyectoBDDlg)
382     ON_EVENT(CProyectoBDDlg, IDC_WINSOCK1, 2 /* ConnectionRequest */, OnConnectionRequestWinsoc
383     ON_EVENT(CProyectoBDDlg, IDC_WINSOCK1, 5 /* Close */, OnCloseWinsock1, VTS_NONE)
384     ON_EVENT(CProyectoBDDlg, IDC_WINSOCK1, 0 /* DataArrival */, OnDataArrivalWinsock1, VTS_I4)
```

```
385     //}}AFX_EVENTSINK_MAP
386 END_EVENTSINK_MAP()
387 
388 void CProyectoBDDlg::OnConnectionRequestWinsock1(long requestID)
389 {
390     // TODO: Add your control notification handler code here
391     m_Socket1.ABROnConnectionRequestWinsock1 (requestID);
392     m_Static1=m_Socket1.ABRConvertirEstado (m_Socket1.GetState ());
393     UpdateData(FALSE);
394 
395     //Enviar Estado Inicial
396     EnviarLeyendaAlSck();
397 //  EnviarEstadoAlSck();
398 }
399 
400 void CProyectoBDDlg::OnCloseWinsock1()
401 {
402     // TODO: Add your control notification handler code here
403     m_Socket1.ABROnCloseWinsock1 ();
404     m_Static1=m_Socket1.ABRConvertirEstado (m_Socket1.GetState ());
405     UpdateData(FALSE);
406     m_Socket1.ABRCerrarSck ();
407     m_Static1=m_Socket1.ABRConvertirEstado (m_Socket1.GetState ());
408     UpdateData(FALSE);
409     m_Socket1.ABREscucharSck ();
410     m_Static1=m_Socket1.ABRConvertirEstado (m_Socket1.GetState ());
411     UpdateData(FALSE);
412 
413 }
414 
415 void CProyectoBDDlg::OnDataArrivalWinsock1(long bytesTotal)
416 {
417     // TODO: Add your control notification handler code here
418     m_Socket1.ABROnDataArrivalWinsock1 (bytesTotal);
419     m_Static1="Recibido: ";
420     m_Static1.Insert(m_Static1.GetLength (),m_Socket1.Dato );
421     UpdateData(FALSE);
422 
423     if (m_Socket1.Dato=="Salida1") OnSalidasSalida1();
424     if (m_Socket1.Dato=="Salida2") OnSalidasSalida2();
425     if (m_Socket1.Dato=="Salida3") OnSalidasSalida3();
426     if (m_Socket1.Dato=="Salida4") OnSalidasSalida4();
427     if (m_Socket1.Dato=="Salida5") OnSalidasSalida5();
428     if (m_Socket1.Dato=="Salida6") OnSalidasSalida6();
429     if (m_Socket1.Dato=="Salida7") OnSalidasSalida7();
430     if (m_Socket1.Dato=="Salida8") OnSalidasSalida8();
431 }
432 
433 
434 void CProyectoBDDlg::OnTimer(UINT nIDEvent)
435 {
436     // TODO: Add your message handler code here and/or call default
437     LeerEntradasPuerto();
438     CDialog::OnTimer(nIDEvent);
439 }
440 
441 void CProyectoBDDlg::OnAccinPropiedades()
442 {
443     // TODO: Add your command handler code here
444     CString BD;
445     BD=AfxGetApp()->GetProfileString ("BD","Path");
446 
447     Propiedades1Dlg.DoModal ();
448     //While (Propiedades1Dlg.IsWindowVisible ())
```

```
449
450     ImprimirLeyenda();
451
452     if (atoi(Leer(&m_EditPort))!= (int) m_Socket1.GetLocalPort ())
453     {
454         m_Socket1.ABRCerrarSck ();
455         m_Socket1.SetLocalPort (atoi(Leer(&m_EditPort)));
456         m_Socket1.ABREscucharSck ();
457         m_Static1=m_Socket1.ABRConvertirEstado (m_Socket1.GetState ());
458         UpdateData(FALSE);
459     }
460     if (BD!=AfxGetApp()->GetProfileString ("BD","Path")) exit(1);
461 }
462
463
464
465 void CProyectoBDDlg::OnAccinCliente()
466 {
467     // TODO: Add your command handler code here
468     ClienteDlg.DoModal ();
469 }
470
471
472
473
```

```
1  // ProyectoBDDlg.h : header file
2  //
3  //{{AFX_INCLUDES()
4  #include "mswinsockcontrol.h"
5  //}}AFX_INCLUDES
6
7  #if !defined(AFX_PROYECTOBDDLG_H__9210F9F6_A9F7_11D5_97E1_FCCD2DB3034F__INCLUDED_)
8  #define AFX_PROYECTOBDDLG_H__9210F9F6_A9F7_11D5_97E1_FCCD2DB3034F__INCLUDED_
9
10 #include "PuertoParalelo.h" // Added by ClassView
11 #include "BaseDatosDlg.h"   // Added by ClassView
12 #include "BaseDatosSet.h"
13
14 #if _MSC_VER > 1000
15 #pragma once
16 #endif // _MSC_VER > 1000
17
18 /////////////////////////////////////////////////////////////////////////////
19 // CProyectoBDDlg dialog
20 class CBaseDatosSet;
21 class CBaseDatosDlg;
22 #include "Propiedades1Dlg.h"    // Added by ClassView
23 #include "ClienteDlg.h" // Added by ClassView
24 class CProyectoBDDlg : public CDialog
25 {
26 // Construction
27 public:
28     void EnviarLeyendaAlSck();
29     CClienteDlg ClienteDlg;
30     void ImprimirLeyenda();
31     CPropiedades1Dlg Propiedades1Dlg;
32     int m_nTimer1;
33     void LeerEntradasPuerto();
34     void EnviarEstadoAlSck();
35     void ActualizarPuerto();
36     CPuertoParalelo PuertoParalelo;
37     void GuardarSalidas();
38     CString Leer(CEdit *Edit);
39     CString MirarEstado(int nSal);
40     void InicializarSalidas();
41     void InsertarRegistro(CString Nombre,CString Estado, CString Hora);
42     CString InsertarHora();
43     void EscribirRegistro ();
44
45     int Salida1;
46     int Salida2;
47     int Salida3;
48     int Salida4;
49     int Salida5;
50     int Salida6;
51     int Salida7;
52     int Salida8;
53     int Entrada1;
54     int Entrada2;
55     int Entrada3;
56     int Entrada4;
57
58     CBaseDatosSet m_BaseDatosDlgSet;
59     CBaseDatosSet* m_pSet;
60
61     CBaseDatosDlg* m_pBaseDatosDlg;
62     CBaseDatosDlg BaseDatosDlg;
63
64     CProyectoBDDlg(CWnd* pParent = NULL);   // standard constructor
```

```
65
66  // Dialog Data
67      //{{AFX_DATA(CProyectoBDDlg)
68      enum { IDD = IDD_PROYECTOBD_DIALOG };
69      CEdit   m_EditPath;
70      CEdit   m_EditPort;
71      CEdit   m_EditIP;
72      CEdit   m_EditEntrada4;
73      CEdit   m_EditEntrada3;
74      CEdit   m_EditEntrada2;
75      CEdit   m_EditEntrada1;
76      CEdit   m_EditSalida8;
77      CEdit   m_EditSalida7;
78      CEdit   m_EditSalida6;
79      CEdit   m_EditSalida5;
80      CEdit   m_EditSalida4;
81      CEdit   m_EditSalida3;
82      CEdit   m_EditSalida2;
83      CEdit   m_EditSalida1;
84      CButton m_CheckEnt4;
85      CButton m_CheckEnt2;
86      CButton m_CheckEnt3;
87      CButton m_CheckEnt1;
88      CButton m_CheckSal8;
89      CButton m_CheckSal7;
90      CButton m_CheckSal6;
91      CButton m_CheckSal5;
92      CButton m_CheckSal4;
93      CButton m_CheckSal3;
94      CButton m_CheckSal2;
95      CButton m_CheckSal1;
96      CEdit   m_EditNombre;
97      CEdit   m_EditHora;
98      CEdit   m_EditEstado;
99      CString m_Static1;
100     CMSWinsockControl   m_Socket1;
101     //}}AFX_DATA
102
103     // ClassWizard generated virtual function overrides
104     //{{AFX_VIRTUAL(CProyectoBDDlg)
105     public:
106     protected:
107     virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV support
108     //}}AFX_VIRTUAL
109
110 // Implementation
111 protected:
112     HICON m_hIcon;
113
114     // Generated message map functions
115     //{{AFX_MSG(CProyectoBDDlg)
116     virtual BOOL OnInitDialog();
117     afx_msg void OnSysCommand(UINT nID, LPARAM lParam);
118     afx_msg void OnPaint();
119     afx_msg HCURSOR OnQueryDragIcon();
120     afx_msg void OnAccinBasededatos();
121     afx_msg void OnSalidasSalida1();
122     afx_msg void OnSalidasSalida2();
123     afx_msg void OnSalidasSalida3();
124     afx_msg void OnSalidasSalida4();
125     afx_msg void OnSalidasSalida5();
126     afx_msg void OnSalidasSalida6();
127     afx_msg void OnSalidasSalida7();
128     afx_msg void OnSalidasSalida8();
```

```
129     afx_msg void OnDestroy();
130     afx_msg void OnCheck1();
131     afx_msg void OnCheck2();
132     afx_msg void OnCheck3();
133     afx_msg void OnCheck4();
134     afx_msg void OnCheck5();
135     afx_msg void OnCheck6();
136     afx_msg void OnCheck7();
137     afx_msg void OnCheck8();
138     afx_msg void OnConnectionRequestWinsock1(long requestID);
139     afx_msg void OnCloseWinsock1();
140     afx_msg void OnDataArrivalWinsock1(long bytesTotal);
141     afx_msg void OnTimer(UINT nIDEvent);
142     afx_msg void OnAccinPropiedades();
143     afx_msg void OnAccinCliente();
144     DECLARE_EVENTSINK_MAP()
145     //}}AFX_MSG
146     DECLARE_MESSAGE_MAP()
147 };
148 
149 //{{AFX_INSERT_LOCATION}}
150 // Microsoft Visual C++ will insert additional declarations immediately before the previous lin
151 
152 #endif // !defined(AFX_PROYECTOBDDLG_H__9210F9F6_A9F7_11D5_97E1_FCCD2DB3034F__INCLUDED_)
153 
```

```
# Microsoft Developer Studio Project File - Name="ProyectoBD" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 6.00
# ** DO NOT EDIT **

# TARGTYPE "Win32 (x86) Application" 0x0101

CFG=ProyectoBD - Win32 Debug
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "ProyectoBD.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "ProyectoBD.mak" CFG="ProyectoBD - Win32 Debug"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "ProyectoBD - Win32 Release" (based on "Win32 (x86) Application")
!MESSAGE "ProyectoBD - Win32 Debug" (based on "Win32 (x86) Application")
!MESSAGE

# Begin Project
# PROP AllowPerConfigDependencies 0
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

!IF  "$(CFG)" == "ProyectoBD - Win32 Release"

# PROP BASE Use_MFC 6
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 6
# PROP Use_Debug_Libraries 0
# PROP Output_Dir "Release"
# PROP Intermediate_Dir "Release"
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MD /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /D "_AFXDLL" /Yu"st
# ADD CPP /nologo /MD /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /D "_AFXDLL" /D "_MBCS"
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /win32
# ADD BASE RSC /l 0xc0a /d "NDEBUG" /d "_AFXDLL"
# ADD RSC /l 0xc0a /d "NDEBUG" /d "_AFXDLL"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 /nologo /subsystem:windows /machine:I386
# ADD LINK32 /nologo /subsystem:windows /machine:I386

!ELSEIF  "$(CFG)" == "ProyectoBD - Win32 Debug"

# PROP BASE Use_MFC 6
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 5
```

```
65 # PROP Use_Debug_Libraries 1
66 # PROP Output_Dir "Debug"
67 # PROP Intermediate_Dir "Debug"
68 # PROP Target_Dir ""
69 # ADD BASE CPP /nologo /MDd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /D "_AFXDL
70 # ADD CPP /nologo /MTd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /D "_MBCS" /Yu"
71 # ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /win32
72 # ADD MTL /nologo /D "_DEBUG" /mktyplib203 /win32
73 # ADD BASE RSC /l 0xc0a /d "_DEBUG" /d "_AFXDLL"
74 # ADD RSC /l 0xc0a /d "_DEBUG"
75 BSC32=bscmake.exe
76 # ADD BASE BSC32 /nologo
77 # ADD BSC32 /nologo
78 LINK32=link.exe
79 # ADD BASE LINK32 /nologo /subsystem:windows /debug /machine:I386 /pdbtype:sept
80 # ADD LINK32 /nologo /subsystem:windows /debug /machine:I386 /pdbtype:sept
81
82 !ENDIF
83
84 # Begin Target
85
86 # Name "ProyectoBD - Win32 Release"
87 # Name "ProyectoBD - Win32 Debug"
88 # Begin Group "Source Files"
89
90 # PROP Default_Filter "cpp;c;cxx;rc;def;r;odl;idl;hpj;bat"
91 # Begin Source File
92
93 SOURCE=.\BaseDatosDlg.cpp
94 # End Source File
95 # Begin Source File
96
97 SOURCE=.\BaseDatosSet.cpp
98 # End Source File
99 # Begin Source File
100
101 SOURCE=.\ClienteDlg.cpp
102 # End Source File
103 # Begin Source File
104
105 SOURCE=.\mswinsockcontrol.cpp
106 # End Source File
107 # Begin Source File
108
109 SOURCE=.\Propiedades1Dlg.cpp
110 # End Source File
111 # Begin Source File
112
113 SOURCE=.\ProyectoBD.cpp
114 # End Source File
115 # Begin Source File
116
117 SOURCE=.\ProyectoBD.rc
118 # End Source File
119 # Begin Source File
120
121 SOURCE=.\ProyectoBDDlg.cpp
122 # End Source File
123 # Begin Source File
124
125 SOURCE=.\PuertoParalelo.cpp
126 # End Source File
127 # Begin Source File
128
```

```
129 SOURCE=.\StdAfx.cpp
130 # ADD CPP /Yc"stdafx.h"
131 # End Source File
132 # End Group
133 # Begin Group "Header Files"
134
135 # PROP Default_Filter "h;hpp;hxx;hm;inl"
136 # Begin Source File
137
138 SOURCE=.\ABRCBaseDatosDlgUtils.h
139 # End Source File
140 # Begin Source File
141
142 SOURCE=.\ABRCClienteDlgUtils.h
143 # End Source File
144 # Begin Source File
145
146 SOURCE=.\ABRCProyectoBDDlgUtils.h
147 # End Source File
148 # Begin Source File
149
150 SOURCE=.\ABRdb.h
151 # End Source File
152 # Begin Source File
153
154 SOURCE=.\ABRWinsock.h
155 # End Source File
156 # Begin Source File
157
158 SOURCE=.\BaseDatosDlg.h
159 # End Source File
160 # Begin Source File
161
162 SOURCE=.\BaseDatosSet.h
163 # End Source File
164 # Begin Source File
165
166 SOURCE=.\ClienteDlg.h
167 # End Source File
168 # Begin Source File
169
170 SOURCE=.\mswinsockcontrol.h
171 # End Source File
172 # Begin Source File
173
174 SOURCE=.\Propiedades1Dlg.h
175 # End Source File
176 # Begin Source File
177
178 SOURCE=.\ProyectoBD.h
179 # End Source File
180 # Begin Source File
181
182 SOURCE=.\ProyectoBDDlg.h
183 # End Source File
184 # Begin Source File
185
186 SOURCE=.\PuertoParalelo.h
187 # End Source File
188 # Begin Source File
189
190 SOURCE=.\Resource.h
191 # End Source File
192 # Begin Source File
```

```
193
194 SOURCE=.\StdAfx.h
195 # End Source File
196 # End Group
197 # Begin Group "Resource Files"
198
199 # PROP Default_Filter "ico;cur;bmp;dlg;rc2;rct;bin;rgs;gif;jpg;jpeg;jpe"
200 # Begin Source File
201
202 SOURCE=.\res\ProyectoBD.ico
203 # End Source File
204 # Begin Source File
205
206 SOURCE=.\res\ProyectoBD.rc2
207 # End Source File
208 # End Group
209 # Begin Source File
210
211 SOURCE=.\ReadMe.txt
212 # End Source File
213 # End Target
214 # End Project
215 # Section ProyectoBD : {248DD896-BB45-11CF-9ABC-0080C7E7B78D}
216 #       2:21:DefaultSinkHeaderFile:mswinsockcontrol.h
217 #       2:16:DefaultSinkClass:CMSWinsockControl
218 # End Section
219 # Section ProyectoBD : {248DD892-BB45-11CF-9ABC-0080C7E7B78D}
220 #       2:5:Class:CMSWinsockControl
221 #       2:10:HeaderFile:mswinsockcontrol.h
222 #       2:8:ImplFile:mswinsockcontrol.cpp
223 # End Section
224
```

```
Microsoft Developer Studio Workspace File, Format Version 6.00
# WARNING: DO NOT EDIT OR DELETE THIS WORKSPACE FILE!

###############################################################################

Project: "ProyectoBD"=.\ProyectoBD.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

###############################################################################

Global:

Package=<5>
{{{
}}}

Package=<3>
{{{
    {248DD896-BB45-11CF-9ABC-0080C7E7B78D}
}}}

###############################################################################


```

```
1  // ProyectoBD.h : main header file for the PROYECTOBD application
2  //
3
4  #if !defined(AFX_PROYECTOBD_H__9210F9F4_A9F7_11D5_97E1_FCCD2DB3034F__INCLUDED_)
5  #define AFX_PROYECTOBD_H__9210F9F4_A9F7_11D5_97E1_FCCD2DB3034F__INCLUDED_
6
7  #if _MSC_VER > 1000
8  #pragma once
9  #endif // _MSC_VER > 1000
10
11 #ifndef __AFXWIN_H__
12     #error include 'stdafx.h' before including this file for PCH
13 #endif
14
15 #include "resource.h"       // main symbols
16
17 /////////////////////////////////////////////////////////////////////////////
18 // CProyectoBDApp:
19 // See ProyectoBD.cpp for the implementation of this class
20 //
21
22 class CProyectoBDApp : public CWinApp
23 {
24 public:
25     CProyectoBDApp();
26
27 // Overrides
28     // ClassWizard generated virtual function overrides
29     //{{AFX_VIRTUAL(CProyectoBDApp)
30     public:
31     virtual BOOL InitInstance();
32     //}}AFX_VIRTUAL
33
34 // Implementation
35
36     //{{AFX_MSG(CProyectoBDApp)
37         // NOTE - the ClassWizard will add and remove member functions here.
38         //    DO NOT EDIT what you see in these blocks of generated code !
39     //}}AFX_MSG
40     DECLARE_MESSAGE_MAP()
41 };
42
43
44 /////////////////////////////////////////////////////////////////////////////
45
46 //{{AFX_INSERT_LOCATION}}
47 // Microsoft Visual C++ will insert additional declarations immediately before the previous line
48
49 #endif // !defined(AFX_PROYECTOBD_H__9210F9F4_A9F7_11D5_97E1_FCCD2DB3034F__INCLUDED_)
50
```

```
ÐÏ à¡± á
Control35=IDC_STATIC,static,1342308352
Control36=IDC_EDIT16,edit,1352730756
Control37=IDC_STATIC,button,1342177287
Control38=IDC_BUTTON_CONECTAR,button,1342242816
Control39=IDC_STATIC,static,1342308352
Control40=IDC_EDIT17,edit,1352730756
Control41=IDC_BUTTON_CERRAR,button,1342242816
Control42=IDC_STATIC,button,1342177287
Control43=IDC_CHK1,button,1342242883
Control44=IDC_CHK2,button,1342242883
Control45=IDC_CHK3,button,1342242883
Control46=IDC_CHK4,button,1342242883
Control47=IDC_CHK5,button,1342242883
Control48=IDC_CHK6,button,1342242883
Control49=IDC_CHK7,button,1342242883
Control50=IDC_CHK8,button,1342242883
Control51=IDC_CHK9,button,1476460611
Control52=IDC_CHK10,button,1476460611
Control53=IDC_CHK11,button,1476460611
Control54=IDC_CHK12,button,1476460611

[CLS:CClienteDlg]
Type=0
HeaderFile=ClienteDlg.h
ImplementationFile=ClienteDlg.cpp
BaseClass=CDialog
Filter=D
VirtualFilter=dWC
LastObject=IDC_WINSOCK1

, 1, 64
Column2=[Estado], 1, 64
Column3=[Hora], 1, 64

[DLG:IDD_PROPIEDADES_DIALOG]
Type=1
Class=CPropiedades1Dlg
ControlCount=36
Control1=IDOK,button,1342242817
Control2=IDCANCEL,button,1342242816
Control3=IDC_STATIC,static,1342308352
Control4=IDC_EDIT1,edit,1350631552
Control5=IDC_STATIC,static,1342308352
Control6=IDC_EDIT2,edit,1350631552
Control7=IDC_STATIC,static,1342308352
Control8=IDC_EDIT3,edit,1350631552
Control9=IDC_STATIC,static,1342308352
Control10=IDC_EDIT4,edit,1350631552
Control11=IDC_STATIC,static,1342308352
Control12=IDC_EDIT5,edit,1350631552
Control13=IDC_STATIC,static,1342308352
Control14=IDC_EDIT6,edit,1350631552
Control15=IDC_STATIC,static,1342308352
Control16=IDC_EDIT7,edit,1350631552
Control17=IDC_STATIC,static,1342308352
Control18=IDC_EDIT8,edit,1350631552
Control19=IDC_STATIC,static,1342308352
Control20=IDC_EDIT9,edit,1350631552
Control21=IDC_STATIC,static,1342308352
Control22=IDC_EDIT10,edit,1350631552
Control23=IDC_STATIC,static,1342308352
Control24=IDC_EDIT11,edit,1350631552
Control25=IDC_STATIC,static,1342308352
```

```
Control26=IDC_EDIT12,edit,1350631552
Control27=IDC_STATIC,static,1342308352
Control28=IDC_EDIT13,edit,1350631552
Control29=IDC_STATIC,static,1342308352
Control30=IDC_EDIT14,edit,1350631552
Control31=IDC_STATIC,button,1342177287
Control32=IDC_STATIC,button,1342177287
Control33=IDC_STATIC,button,1342177287
Control34=IDC_EDIT15,edit,1350631552
Control35=IDC_STATIC,static,1342308352
Control36=IDC_STATIC,button,1342177287

[CLS:CPropiedades1Dlg]
Type=0
HeaderFile=Propiedades1Dlg.h
ImplementationFile=Propiedades1Dlg.cpp
BaseClass=CDialog
Filter=D
LastObject=CPropiedades1Dlg
VirtualFilter=dWC

[DLG:IDD_CLIENTEDLG_DIALOG]
Type=1
Class=CClienteDlg
ControlCount=54
Control1=IDC_STATIC1,static,1342308352
Control2=IDC_STATIC,button,1342177287
Control3=IDC_WINSOCK1,{248DD896-BB45-11CF-9ABC-0080C7E7B78D},1342242816
Control4=IDC_STATIC,button,1342177287
Control5=IDC_STATIC,static,1342308352
Control6=IDC_EDIT01,edit,1350633600
Control7=IDC_STATIC,static,1342308352
Control8=IDC_EDIT02,edit,1350633600
Control9=IDC_STATIC,static,1342308352
Control10=IDC_EDIT03,edit,1350633600
Control11=IDC_STATIC,static,1342308352
Control12=IDC_EDIT04,edit,1350633600
Control13=IDC_STATIC,static,1342308352
Control14=IDC_EDIT05,edit,1350633600
Control15=IDC_STATIC,static,1342308352
Control16=IDC_EDIT06,edit,1350633600
Control17=IDC_STATIC,static,1342308352
Control18=IDC_EDIT07,edit,1350633600
Control19=IDC_STATIC,static,1342308352
Control20=IDC_EDIT08,edit,1350633600
Control21=IDC_STATIC,static,1342308352
Control22=IDC_EDIT09,edit,1350633600
Control23=IDC_STATIC,static,1342308352
Control24=IDC_EDIT010,edit,1350633600
Control25=IDC_STATIC,static,1342308352
Control26=IDC_EDIT011,edit,1350633600
Control27=IDC_STATIC,static,1342308352
Control28=IDC_EDIT012,edit,1350633600
Control29=IDC_STATIC,static,1342308352
Control30=IDC_EDIT013,edit,1350631552
Control31=IDC_STATIC,static,1342308352
Control32=IDC_EDIT014,edit,1Øãx
Control35=IDC_STATIC,static,1342308352
Control36=IDC_EDIT16,edit,1352730756
Control37=IDC_STATIC,button,1342177287
Control38=IDC_BUTTON_CONECTAR,button,1342242816
Control39=IDC_STATIC,static,1342308352
Control40=IDC_EDIT17,edit,1352730756
Control41=IDC_BUTTON_CERRAR,button,1342242816
```

```
Control42=IDC_STATIC,button,1342177287
Control43=IDC_CHK1,button,1342242883
Control44=IDC_CHK2,button,1342242883
Control45=IDC_CHK3,button,1342242883
Control46=IDC_CHK4,button,1342242883
Control47=IDC_CHK5,button,1342242883
Control48=IDC_CHK6,button,1342242883
Control49=IDC_CHK7,button,1342242883
Control50=IDC_CHK8,button,1342242883
Control51=IDC_CHK9,button,1476460611
Control52=IDC_CHK10,button,1476460611
Control53=IDC_CHK11,button,1476460611
Control54=IDC_CHK12,button,1476460611

[CLS:CClienteDlg]
Type=0
HeaderFile=ClienteDlg.h
ImplementationFile=ClienteDlg.cpp
BaseClass=CDialog
Filter=D
VirtualFilter=dWC
LastObject=IDC_WINSOCK1

, 1, 64
Column2=[Estado], 1, 64
Column3=[Hora], 1, 64

[DLG:IDD_PROPIEDADES_DIALOG]
Type=1
Class=CPropiedades1Dlg
ControlCount=36
Control1=IDOK,button,1342242817
Control2=IDCANCEL,button,1342242816
Control3=IDC_STATIC,static,1342308352
Control4=IDC_EDIT1,edit,1350631552
Control5=IDC_STATIC,static,1342308352
Control6=IDC_EDIT2,edit,1350631552
Control7=IDC_STATIC,static,1342308352
Control8=IDC_EDIT3,edit,1350631552
Control9=IDC_STATIC,static,1342308352
Control10=IDC_EDIT4,edit,1350631552
Control11=IDC_STATIC,static,1342308352
Control12=IDC_EDIT5,edit,1350631552
Control13=IDC_STATIC,static,1342308352
Control14=IDC_EDIT6,edit,1350631552
Control15=IDC_STATIC,static,1342308352
Control16=IDC_EDIT7,edit,1350631552
Control17=IDC_STATIC,static,1342308352
Control18=IDC_EDIT8,edit,1350631552
Control19=IDC_STATIC,static,1342308352
Control20=IDC_EDIT9,edit,1350631552
Control21=IDC_STATIC,static,1342308352
Control22=IDC_EDIT10,edit,1350631552
Control23=IDC_STATIC,static,1342308352
Control24=IDC_EDIT11,edit,1350631552
Control25=IDC_STATIC,static,1342308352
Control26=IDC_EDIT12,edit,1350631552
Control27=IDC_STATIC,static,1342308352
Control28=IDC_EDIT13,edit,1350631552
Control29=IDC_STATIC,static,1342308352
Control30=IDC_EDIT14,edit,1350631552
Control31=IDC_STATIC,button,1342177287
Control32=IDC_STATIC,button,1342177287
Control33=IDC_STATIC,button,1342177287
```

```
Control34=IDC_EDIT15,edit,1350631552
Control35=IDC_STATIC,static,1342308352
Control36=IDC_STATIC,button,1342177287

[CLS:CPropiedades1Dlg]
Type=0
HeaderFile=Propiedades1Dlg.h
ImplementationFile=Propiedades1Dlg.cpp
BaseClass=CDialog
Filter=D
LastObject=CPropiedades1Dlg
VirtualFilter=dWC

[DLG:IDD_CLIENTEDLG_DIALOG]
Type=1
Class=CClienteDlg
ControlCount=54
Control1=IDC_STATIC1,static,1342308352
Control2=IDC_STATIC,button,1342177287
Control3=IDC_WINSOCK1,{248DD896-BB45-11CF-9ABC-0080C7E7B78D},1342242816
Control4=IDC_STATIC,button,1342177287
Control5=IDC_STATIC,static,1342308352
Control6=IDC_EDIT01,edit,1350633600
Control7=IDC_STATIC,static,1342308352
Control8=IDC_EDIT02,edit,1350633600
Control9=IDC_STATIC,static,1342308352
Control10=IDC_EDIT03,edit,1350633600
Control11=IDC_STATIC,static,1342308352
Control12=IDC_EDIT04,edit,1350633600
Control13=IDC_STATIC,static,1342308352
Control14=IDC_EDIT05,edit,1350633600
Control15=IDC_STATIC,static,1342308352
Control16=IDC_EDIT06,edit,1350633600
Control17=IDC_STATIC,static,1342308352
Control18=IDC_EDIT07,edit,1350633600
Control19=IDC_STATIC,static,1342308352
Control20=IDC_EDIT08,edit,1350633600
Control21=IDC_STATIC,static,1342308352
Control22=IDC_EDIT09,edit,1350633600
Control23=IDC_STATIC,static,1342308352
Control24=IDC_EDIT010,edit,1350633600
Control25=IDC_STATIC,static,1342308352
Control26=IDC_EDIT011,edit,1350633600
Control27=IDC_STATIC,static,1342308352
Control28=IDC_EDIT012,edit,1350633600
Control29=IDC_STATIC,static,1342308352
Control30=IDC_EDIT013,edit,1350631552
Control31=IDC_STATIC,static,1342308352
Control32=IDC_EDIT014,edit,1Øãx
Control35=IDC_STATIC,static,1342308352
Control36=IDC_EDIT16,edit,1352730756
Control37=IDC_STATIC,button,1342177287
Control38=IDC_BUTTON_CONECTAR,button,1342242816
Control39=IDC_STATIC,static,1342308352
Control40=IDC_EDIT17,edit,1352730756
Control41=IDC_BUTTON_CERRAR,button,1342242816
Control42=IDC_STATIC,button,1342177287
Control43=IDC_CHK1,button,1342242883
Control44=IDC_CHK2,button,1342242883
Control45=IDC_CHK3,button,1342242883
Control46=IDC_CHK4,button,1342242883
Control47=IDC_CHK5,button,1342242883
Control48=IDC_CHK6,button,1342242883
Control49=IDC_CHK7,button,1342242883
```

```
Control50=IDC_CHK8,button,1342242883
Control51=IDC_CHK9,button,1476460611
Control52=IDC_CHK10,button,1476460611
Control53=IDC_CHK11,button,1476460611
Control54=IDC_CHK12,button,1476460611

[CLS:CClienteDlg]
Type=0
HeaderFile=ClienteDlg.h
ImplementationFile=ClienteDlg.cpp
BaseClass=CDialog
Filter=D
VirtualFilter=dWC
LastObject=IDC_WINSOCK1

, 1, 64
Column2=[Estado], 1, 64
Column3=[Hora], 1, 64

[DLG:IDD_PROPIEDADES_DIALOG]
Type=1
Class=CPropiedades1Dlg
ControlCount=36
Control1=IDOK,button,1342242817
Control2=IDCANCEL,button,1342242816
Control3=IDC_STATIC,static,1342308352
Control4=IDC_EDIT1,edit,1350631552
Control5=IDC_STATIC,static,1342308352
Control6=IDC_EDIT2,edit,1350631552
Control7=IDC_STATIC,static,1342308352
Control8=IDC_EDIT3,edit,1350631552
Control9=IDC_STATIC,static,1342308352
Control10=IDC_EDIT4,edit,1350631552
Control11=IDC_STATIC,static,1342308352
Control12=IDC_EDIT5,edit,1350631552
Control13=IDC_STATIC,static,1342308352
Control14=IDC_EDIT6,edit,1350631552
Control15=IDC_STATIC,static,1342308352
Control16=IDC_EDIT7,edit,1350631552
Control17=IDC_STATIC,static,1342308352
Control18=IDC_EDIT8,edit,1350631552
Control19=IDC_STATIC,static,1342308352
Control20=IDC_EDIT9,edit,1350631552
Control21=IDC_STATIC,static,1342308352
Control22=IDC_EDIT10,edit,1350631552
Control23=IDC_STATIC,static,1342308352
Control24=IDC_EDIT11,edit,1350631552
Control25=IDC_STATIC,static,1342308352
Control26=IDC_EDIT12,edit,1350631552
Control27=IDC_STATIC,static,1342308352
Control28=IDC_EDIT13,edit,1350631552
Control29=IDC_STATIC,static,1342308352
Control30=IDC_EDIT14,edit,1350631552
Control31=IDC_STATIC,button,1342177287
Control32=IDC_STATIC,button,1342177287
Control33=IDC_STATIC,button,1342177287
Control34=IDC_EDIT15,edit,1350631552
Control35=IDC_STATIC,static,1342308352
Control36=IDC_STATIC,button,1342177287

[CLS:CPropiedades1Dlg]
Type=0
HeaderFile=Propiedades1Dlg.h
ImplementationFile=Propiedades1Dlg.cpp
```

```
BaseClass=CDialog
Filter=D
LastObject=CPropiedades1Dlg
VirtualFilter=dWC

[DLG:IDD_CLIENTEDLG_DIALOG]
Type=1
Class=CClienteDlg
ControlCount=54
Control1=IDC_STATIC1,static,1342308352
Control2=IDC_STATIC,button,1342177287
Control3=IDC_WINSOCK1,{248DD896-BB45-11CF-9ABC-0080C7E7B78D},1342242816
Control4=IDC_STATIC,button,1342177287
Control5=IDC_STATIC,static,1342308352
Control6=IDC_EDIT01,edit,1350633600
Control7=IDC_STATIC,static,1342308352
Control8=IDC_EDIT02,edit,1350633600
Control9=IDC_STATIC,static,1342308352
Control10=IDC_EDIT03,edit,1350633600
Control11=IDC_STATIC,static,1342308352
Control12=IDC_EDIT04,edit,1350633600
Control13=IDC_STATIC,static,1342308352
Control14=IDC_EDIT05,edit,1350633600
Control15=IDC_STATIC,static,1342308352
Control16=IDC_EDIT06,edit,1350633600
Control17=IDC_STATIC,static,1342308352
Control18=IDC_EDIT07,edit,1350633600
Control19=IDC_STATIC,static,1342308352
Control20=IDC_EDIT08,edit,1350633600
Control21=IDC_STATIC,static,1342308352
Control22=IDC_EDIT09,edit,1350633600
Control23=IDC_STATIC,static,1342308352
Control24=IDC_EDIT010,edit,1350633600
Control25=IDC_STATIC,static,1342308352
Control26=IDC_EDIT011,edit,1350633600
Control27=IDC_STATIC,static,1342308352
Control28=IDC_EDIT012,edit,1350633600
Control29=IDC_STATIC,static,1342308352
Control30=IDC_EDIT013,edit,1350631552
Control31=IDC_STATIC,static,1342308352
Control32=IDC_EDIT014,edit,1Øãx
Control42=IDC_STATIC,button,1342177287
Control43=IDC_CHK1,button,1342242883
Control44=IDC_CHK2,button,1342242883
Control45=IDC_CHK3,button,1342242883
Control46=IDC_CHK4,button,1342242883
Control47=IDC_CHK5,button,1342242883
Control48=IDC_CHK6,button,1342242883
Control49=IDC_CHK7,button,1342242883
Control50=IDC_CHK8,button,1342242883
Control51=IDC_CHK9,button,1476460611
Control52=IDC_CHK10,button,1476460611
Control53=IDC_CHK11,button,1476460611
Control54=IDC_CHK12,button,1476460611

[CLS:CClienteDlg]
Type=0
HeaderFile=ClienteDlg.h
ImplementationFile=ClienteDlg.cpp
BaseClass=CDialog
Filter=D
VirtualFilter=dWC
LastObject=IDC_WINSOCK1

```

```
, 1, 64
Column2=[Estado], 1, 64
Column3=[Hora], 1, 64

[DLG:IDD_PROPIEDADES_DIALOG]
Type=1
Class=CPropiedades1Dlg
ControlCount=36
Control1=IDOK,button,1342242817
Control2=IDCANCEL,button,1342242816
Control3=IDC_STATIC,static,1342308352
Control4=IDC_EDIT1,edit,1350631552
Control5=IDC_STATIC,static,1342308352
Control6=IDC_EDIT2,edit,1350631552
Control7=IDC_STATIC,static,1342308352
Control8=IDC_EDIT3,edit,1350631552
Control9=IDC_STATIC,static,1342308352
Control10=IDC_EDIT4,edit,1350631552
Control11=IDC_STATIC,static,1342308352
Control12=IDC_EDIT5,edit,1350631552
Control13=IDC_STATIC,static,1342308352
Control14=IDC_EDIT6,edit,1350631552
Control15=IDC_STATIC,static,1342308352
Control16=IDC_EDIT7,edit,1350631552
Control17=IDC_STATIC,static,1342308352
Control18=IDC_EDIT8,edit,1350631552
Control19=IDC_STATIC,static,1342308352
Control20=IDC_EDIT9,edit,1350631552
Control21=IDC_STATIC,static,1342308352
Control22=IDC_EDIT10,edit,1350631552
Control23=IDC_STATIC,static,1342308352
Control24=IDC_EDIT11,edit,1350631552
Control25=IDC_STATIC,static,1342308352
Control26=IDC_EDIT12,edit,1350631552
Control27=IDC_STATIC,static,1342308352
Control28=IDC_EDIT13,edit,1350631552
Control29=IDC_STATIC,static,1342308352
Control30=IDC_EDIT14,edit,1350631552
Control31=IDC_STATIC,button,1342177287
Control32=IDC_STATIC,button,1342177287
Control33=IDC_STATIC,button,1342177287
Control34=IDC_EDIT15,edit,1350631552
Control35=IDC_STATIC,static,1342308352
Control36=IDC_STATIC,button,1342177287

[CLS:CPropiedades1Dlg]
Type=0
HeaderFile=Propiedades1Dlg.h
ImplementationFile=Propiedades1Dlg.cpp
BaseClass=CDialog
Filter=D
LastObject=CPropiedades1Dlg
VirtualFilter=dWC

[DLG:IDD_CLIENTEDLG_DIALOG]
Type=1
Class=CClienteDlg
ControlCount=54
Control1=IDC_STATIC1,static,1342308352
Control2=IDC_STATIC,button,1342177287
Control3=IDC_WINSOCK1,{248DD896-BB45-11CF-9ABC-0080C7E7B78D},1342242816
Control4=IDC_STATIC,button,1342177287
Control5=IDC_STATIC,static,1342308352
Control6=IDC_EDIT01,edit,1350633600
```

```
Control7=IDC_STATIC,static,1342308352
Control8=IDC_EDIT02,edit,1350633600
Control9=IDC_STATIC,static,1342308352
Control10=IDC_EDIT03,edit,1350633600
Control11=IDC_STATIC,static,1342308352
Control12=IDC_EDIT04,edit,1350633600
Control13=IDC_STATIC,static,1342308352
Control14=IDC_EDIT05,edit,1350633600
Control15=IDC_STATIC,static,1342308352
Control16=IDC_EDIT06,edit,1350633600
Control17=IDC_STATIC,static,1342308352
Control18=IDC_EDIT07,edit,1350633600
Control19=IDC_STATIC,static,1342308352
Control20=IDC_EDIT08,edit,1350633600
Control21=IDC_STATIC,static,1342308352
Control22=IDC_EDIT09,edit,1350633600
Control23=IDC_STATIC,static,1342308352
Control24=IDC_EDIT010,edit,1350633600
Control25=IDC_STATIC,static,1342308352
Control26=IDC_EDIT011,edit,1350633600
Control27=IDC_STATIC,static,1342308352
Control28=IDC_EDIT012,edit,1350633600
Control29=IDC_STATIC,static,1342308352
Control30=IDC_EDIT013,edit,1350631552
Control31=IDC_STATIC,static,1342308352
Control32=IDC_EDIT014,edit,1Øãx
Control42=IDC_STATIC,button,1342177287
Control43=IDC_CHK1,button,1342242883
Control44=IDC_CHK2,button,1342242883
Control45=IDC_CHK3,button,1342242883
Control46=IDC_CHK4,button,1342242883
Control47=IDC_CHK5,button,1342242883
Control48=IDC_CHK6,button,1342242883
Control49=IDC_CHK7,button,1342242883
Control50=IDC_CHK8,button,1342242883
Control51=IDC_CHK9,button,1476460611
Control52=IDC_CHK10,button,1476460611
Control53=IDC_CHK11,button,1476460611
Control54=IDC_CHK12,button,1476460611

[CLS:CClienteDlg]
Type=0
HeaderFile=ClienteDlg.h
ImplementationFile=ClienteDlg.cpp
BaseClass=CDialog
Filter=D
VirtualFilter=dWC
LastObject=IDC_WINSOCK1

, 1, 64
Column2=[Estado], 1, 64
Column3=[Hora], 1, 64

[DLG:IDD_PROPIEDADES_DIALOG]
Type=1
Class=CPropiedades1Dlg
ControlCount=36
Control1=IDOK,button,1342242817
Control2=IDCANCEL,button,1342242816
Control3=IDC_STATIC,static,1342308352
Control4=IDC_EDIT1,edit,1350631552
Control5=IDC_STATIC,static,1342308352
Control6=IDC_EDIT2,edit,1350631552
Control7=IDC_STATIC,static,1342308352
```

```
Control8=IDC_EDIT3,edit,1350631552
Control9=IDC_STATIC,static,1342308352
Control10=IDC_EDIT4,edit,1350631552
Control11=IDC_STATIC,static,1342308352
Control12=IDC_EDIT5,edit,1350631552
Control13=IDC_STATIC,static,1342308352
Control14=IDC_EDIT6,edit,1350631552
Control15=IDC_STATIC,static,1342308352
Control16=IDC_EDIT7,edit,1350631552
Control17=IDC_STATIC,static,1342308352
Control18=IDC_EDIT8,edit,1350631552
Control19=IDC_STATIC,static,1342308352
Control20=IDC_EDIT9,edit,1350631552
Control21=IDC_STATIC,static,1342308352
Control22=IDC_EDIT10,edit,1350631552
Control23=IDC_STATIC,static,1342308352
Control24=IDC_EDIT11,edit,1350631552
Control25=IDC_STATIC,static,1342308352
Control26=IDC_EDIT12,edit,1350631552
Control27=IDC_STATIC,static,1342308352
Control28=IDC_EDIT13,edit,1350631552
Control29=IDC_STATIC,static,1342308352
Control30=IDC_EDIT14,edit,1350631552
Control31=IDC_STATIC,button,1342177287
Control32=IDC_STATIC,button,1342177287
Control33=IDC_STATIC,button,1342177287
Control34=IDC_EDIT15,edit,1350631552
Control35=IDC_STATIC,static,1342308352
Control36=IDC_STATIC,button,1342177287

[CLS:CPropiedades1Dlg]
Type=0
HeaderFile=Propiedades1Dlg.h
ImplementationFile=Propiedades1Dlg.cpp
BaseClass=CDialog
Filter=D
LastObject=CPropiedades1Dlg
VirtualFilter=dWC

[DLG:IDD_CLIENTEDLG_DIALOG]
Type=1
Class=CClienteDlg
ControlCount=54
Control1=IDC_STATIC1,static,1342308352
Control2=IDC_STATIC,button,1342177287
Control3=IDC_WINSOCK1,{248DD896-BB45-11CF-9ABC-0080C7E7B78D},1342242816
Control4=IDC_STATIC,button,1342177287
Control5=IDC_STATIC,static,1342308352
Control6=IDC_EDIT01,edit,1350633600
Control7=IDC_STATIC,static,1342308352
Control8=IDC_EDIT02,edit,1350633600
Control9=IDC_STATIC,static,1342308352
Control10=IDC_EDIT03,edit,1350633600
Control11=IDC_STATIC,static,1342308352
Control12=IDC_EDIT04,edit,1350633600
Control13=IDC_STATIC,static,1342308352
Control14=IDC_EDIT05,edit,1350633600
Control15=IDC_STATIC,static,1342308352
Control16=IDC_EDIT06,edit,1350633600
Control17=IDC_STATIC,static,1342308352
Control18=IDC_EDIT07,edit,1350633600
Control19=IDC_STATIC,static,1342308352
Control20=IDC_EDIT08,edit,1350633600
Control21=IDC_STATIC,static,1342308352
```

```
Control22=IDC_EDIT09,edit,1350633600
Control23=IDC_STATIC,static,1342308352
Control24=IDC_EDIT010,edit,1350633600
Control25=IDC_STATIC,static,1342308352
Control26=IDC_EDIT011,edit,1350633600
Control27=IDC_STATIC,static,1342308352
Control28=IDC_EDIT012,edit,1350633600
Control29=IDC_STATIC,static,1342308352
Control30=IDC_EDIT013,edit,1350631552
Control31=IDC_STATIC,static,1342308352
Control32=IDC_EDIT014,edit,1Øãx
Control45=IDC_CHK3,button,1342242883
Control46=IDC_CHK4,button,1342242883
Control47=IDC_CHK5,button,1342242883
Control48=IDC_CHK6,button,1342242883
Control49=IDC_CHK7,button,1342242883
Control50=IDC_CHK8,button,1342242883
Control51=IDC_CHK9,button,1476460611
Control52=IDC_CHK10,button,1476460611
Control53=IDC_CHK11,button,1476460611
Control54=IDC_CHK12,button,1476460611

[CLS:CClienteDlg]
Type=0
HeaderFile=ClienteDlg.h
ImplementationFile=ClienteDlg.cpp
BaseClass=CDialog
Filter=D
VirtualFilter=dWC
LastObject=IDC_WINSOCK1

, 1, 64
Column2=[Estado], 1, 64
Column3=[Hora], 1, 64

[DLG:IDD_PROPIEDADES_DIALOG]
Type=1
Class=CPropiedades1Dlg
ControlCount=36
Control1=IDOK,button,1342242817
Control2=IDCANCEL,button,1342242816
Control3=IDC_STATIC,static,1342308352
Control4=IDC_EDIT1,edit,1350631552
Control5=IDC_STATIC,static,1342308352
Control6=IDC_EDIT2,edit,1350631552
Control7=IDC_STATIC,static,1342308352
Control8=IDC_EDIT3,edit,1350631552
Control9=IDC_STATIC,static,1342308352
Control10=IDC_EDIT4,edit,1350631552
Control11=IDC_STATIC,static,1342308352
Control12=IDC_EDIT5,edit,1350631552
Control13=IDC_STATIC,static,1342308352
Control14=IDC_EDIT6,edit,1350631552
Control15=IDC_STATIC,static,1342308352
Control16=IDC_EDIT7,edit,1350631552
Control17=IDC_STATIC,static,1342308352
Control18=IDC_EDIT8,edit,1350631552
Control19=IDC_STATIC,static,1342308352
Control20=IDC_EDIT9,edit,1350631552
Control21=IDC_STATIC,static,1342308352
Control22=IDC_EDIT10,edit,1350631552
Control23=IDC_STATIC,static,1342308352
Control24=IDC_EDIT11,edit,1350631552
Control25=IDC_STATIC,static,1342308352
```

```
Control26=IDC_EDIT12,edit,1350631552
Control27=IDC_STATIC,static,1342308352
Control28=IDC_EDIT13,edit,1350631552
Control29=IDC_STATIC,static,1342308352
Control30=IDC_EDIT14,edit,1350631552
Control31=IDC_STATIC,button,1342177287
Control32=IDC_STATIC,button,1342177287
Control33=IDC_STATIC,button,1342177287
Control34=IDC_EDIT15,edit,1350631552
Control35=IDC_STATIC,static,1342308352
Control36=IDC_STATIC,button,1342177287

[CLS:CPropiedades1Dlg]
Type=0
HeaderFile=Propiedades1Dlg.h
ImplementationFile=Propiedades1Dlg.cpp
BaseClass=CDialog
Filter=D
LastObject=CPropiedades1Dlg
VirtualFilter=dWC

[DLG:IDD_CLIENTEDLG_DIALOG]
Type=1
Class=CClienteDlg
ControlCount=54
Control1=IDC_STATIC1,static,1342308352
Control2=IDC_STATIC,button,1342177287
Control3=IDC_WINSOCK1,{248DD896-BB45-11CF-9ABC-0080C7E7B78D},1342242816
Control4=IDC_STATIC,button,1342177287
Control5=IDC_STATIC,static,1342308352
Control6=IDC_EDIT01,edit,1350633600
Control7=IDC_STATIC,static,1342308352
Control8=IDC_EDIT02,edit,1350633600
Control9=IDC_STATIC,static,1342308352
Control10=IDC_EDIT03,edit,1350633600
Control11=IDC_STATIC,static,1342308352
Control12=IDC_EDIT04,edit,1350633600
Control13=IDC_STATIC,static,1342308352
Control14=IDC_EDIT05,edit,1350633600
Control15=IDC_STATIC,static,1342308352
Control16=IDC_EDIT06,edit,1350633600
Control17=IDC_STATIC,static,1342308352
Control18=IDC_EDIT07,edit,1350633600
Control19=IDC_STATIC,static,1342308352
Control20=IDC_EDIT08,edit,1350633600
Control21=IDC_STATIC,static,1342308352
Control22=IDC_EDIT09,edit,1350633600
Control23=IDC_STATIC,static,1342308352
Control24=IDC_EDIT010,edit,1350633600
Control25=IDC_STATIC,static,1342308352
Control26=IDC_EDIT011,edit,1350633600
Control27=IDC_STATIC,static,1342308352
Control28=IDC_EDIT012,edit,1350633600
Control29=IDC_STATIC,static,1342308352
Control30=IDC_EDIT013,edit,1350631552
Control31=IDC_STATIC,static,1342308352
Control32=IDC_EDIT014,edit,1Øãx
Control45=IDC_CHK3,button,1342242883
Control46=IDC_CHK4,button,1342242883
Control47=IDC_CHK5,button,1342242883
Control48=IDC_CHK6,button,1342242883
Control49=IDC_CHK7,button,1342242883
Control50=IDC_CHK8,button,1342242883
Control51=IDC_CHK9,button,1476460611
```

```
Control52=IDC_CHK10,button,1476460611
Control53=IDC_CHK11,button,1476460611
Control54=IDC_CHK12,button,1476460611

[CLS:CClienteDlg]
Type=0
HeaderFile=ClienteDlg.h
ImplementationFile=ClienteDlg.cpp
BaseClass=CDialog
Filter=D
VirtualFilter=dWC
LastObject=IDC_WINSOCK1

, 1, 64
Column2=[Estado], 1, 64
Column3=[Hora], 1, 64

[DLG:IDD_PROPIEDADES_DIALOG]
Type=1
Class=CPropiedades1Dlg
ControlCount=36
Control1=IDOK,button,1342242817
Control2=IDCANCEL,button,1342242816
Control3=IDC_STATIC,static,1342308352
Control4=IDC_EDIT1,edit,1350631552
Control5=IDC_STATIC,static,1342308352
Control6=IDC_EDIT2,edit,1350631552
Control7=IDC_STATIC,static,1342308352
Control8=IDC_EDIT3,edit,1350631552
Control9=IDC_STATIC,static,1342308352
Control10=IDC_EDIT4,edit,1350631552
Control11=IDC_STATIC,static,1342308352
Control12=IDC_EDIT5,edit,1350631552
Control13=IDC_STATIC,static,1342308352
Control14=IDC_EDIT6,edit,1350631552
Control15=IDC_STATIC,static,1342308352
Control16=IDC_EDIT7,edit,1350631552
Control17=IDC_STATIC,static,1342308352
Control18=IDC_EDIT8,edit,1350631552
Control19=IDC_STATIC,static,1342308352
Control20=IDC_EDIT9,edit,1350631552
Control21=IDC_STATIC,static,1342308352
Control22=IDC_EDIT10,edit,1350631552
Control23=IDC_STATIC,static,1342308352
Control24=IDC_EDIT11,edit,1350631552
Control25=IDC_STATIC,static,1342308352
Control26=IDC_EDIT12,edit,1350631552
Control27=IDC_STATIC,static,1342308352
Control28=IDC_EDIT13,edit,1350631552
Control29=IDC_STATIC,static,1342308352
Control30=IDC_EDIT14,edit,1350631552
Control31=IDC_STATIC,button,1342177287
Control32=IDC_STATIC,button,1342177287
Control33=IDC_STATIC,button,1342177287
Control34=IDC_EDIT15,edit,1350631552
Control35=IDC_STATIC,static,1342308352
Control36=IDC_STATIC,button,1342177287

[CLS:CPropiedades1Dlg]
Type=0
HeaderFile=Propiedades1Dlg.h
ImplementationFile=Propiedades1Dlg.cpp
BaseClass=CDialog
Filter=D
```

```
Control45=IDC_CHK3,button,1342242883
Control46=IDC_CHK4,button,1342242883
Control47=IDC_CHK5,button,1342242883
Control48=IDC_CHK6,button,1342242883
Control49=IDC_CHK7,button,1342242883
Control50=IDC_CHK8,button,1342242883
Control51=IDC_CHK9,button,1476460611
Control52=IDC_CHK10,button,1476460611
Control53=IDC_CHK11,button,1476460611
Control54=IDC_CHK12,button,1476460611

[CLS:CClienteDlg]
Type=0
HeaderFile=ClienteDlg.h
ImplementationFile=ClienteDlg.cpp
BaseClass=CDialog
Filter=D
VirtualFilter=dWC
LastObject=IDC_WINSOCK1

, 1, 64
Column2=[Estado], 1, 64
Column3=[Hora], 1, 64

[DLG:IDD_PROPIEDADES_DIALOG]
```

```
Type=1
Class=CPropiedades1Dlg
ControlCount=36
Control1=IDOK,button,1342242817
Control2=IDCANCEL,button,1342242816
Control3=IDC_STATIC,static,1342308352
Control4=IDC_EDIT1,edit,1350631552
Control5=IDC_STATIC,static,1342308352
Control6=IDC_EDIT2,edit,1350631552
Control7=IDC_STATIC,static,1342308352
Control8=IDC_EDIT3,edit,1350631552
Control9=IDC_STATIC,static,1342308352
Control10=IDC_EDIT4,edit,1350631552
Control11=IDC_STATIC,static,1342308352
Control12=IDC_EDIT5,edit,1350631552
Control13=IDC_STATIC,static,1342308352
Control14=IDC_EDIT6,edit,1350631552
Control15=IDC_STATIC,static,1342308352
Control16=IDC_EDIT7,edit,1350631552
Control17=IDC_STATIC,static,1342308352
Control18=IDC_EDIT8,edit,1350631552
Control19=IDC_STATIC,static,1342308352
Control20=IDC_EDIT9,edit,1350631552
Control21=IDC_STATIC,static,1342308352
Control22=IDC_EDIT10,edit,1350631552
Control23=IDC_STATIC,static,1342308352
Control24=IDC_EDIT11,edit,1350631552
Control25=IDC_STATIC,static,1342308352
Control26=IDC_EDIT12,edit,1350631552
Control27=IDC_STATIC,static,1342308352
Control28=IDC_EDIT13,edit,1350631552
Control29=IDC_STATIC,static,1342308352
Control30=IDC_EDIT14,edit,1350631552
Control31=IDC_STATIC,button,1342177287
Control32=IDC_STATIC,button,1342177287
Control33=IDC_STATIC,button,1342177287
Control34=IDC_EDIT15,edit,1350631552
Control35=IDC_STATIC,static,1342308352
Control36=IDC_STATIC,button,1342177287

[CLS:CPropiedades1Dlg]
Type=0
HeaderFile=Propiedades1Dlg.h
ImplementationFile=Propiedades1Dlg.cpp
BaseClass=CDialog
Filter=D
LastObject=CPropiedades1Dlg
VirtualFilter=dWC

[DLG:IDD_CLIENTEDLG_DIALOG]
Type=1
Class=CClienteDlg
ControlCount=54
Control1=IDC_STATIC1,static,1342308352
Control2=IDC_STATIC,button,1342177287
Control3=IDC_WINSOCK1,{248DD896-BB45-11CF-9ABC-0080C7E7B78D},1342242816
Control4=IDC_STATIC,button,1342177287
Control5=IDC_STATIC,static,1342308352
Control6=IDC_EDIT01,edit,1350633600
Control7=IDC_STATIC,static,1342308352
Control8=IDC_EDIT02,edit,1350633600
Control9=IDC_STATIC,static,1342308352
Control10=IDC_EDIT03,edit,1350633600
Control11=IDC_STATIC,static,1342308352
```

```
Control12=IDC_EDIT04,edit,1350633600
Control13=IDC_STATIC,static,1342308352
Control14=IDC_EDIT05,edit,1350633600
Control15=IDC_STATIC,static,1342308352
Control16=IDC_EDIT06,edit,1350633600
Control17=IDC_STATIC,static,1342308352
Control18=IDC_EDIT07,edit,1350633600
Control19=IDC_STATIC,static,1342308352
Control20=IDC_EDIT08,edit,1350633600
Control21=IDC_STATIC,static,1342308352
Control22=IDC_EDIT09,edit,1350633600
Control23=IDC_STATIC,static,1342308352
Control24=IDC_EDIT010,edit,1350633600
Control25=IDC_STATIC,static,1342308352
Control26=IDC_EDIT011,edit,1350633600
Control27=IDC_STATIC,static,1342308352
Control28=IDC_EDIT012,edit,1350633600
Control29=IDC_STATIC,static,1342308352
Control30=IDC_EDIT013,edit,1350631552
Control31=IDC_STATIC,static,1342308352
Control32=IDC_EDIT014,edit,1Øãx         Control-C
HeaderFile=ClienteDlg.h
ImplementationFile=ClienteDlg.cpp
BaseClass=CDialog
Filter=D
VirtualFilter=dWC
LastObject=IDC_WINSOCK1

, 1, 64
Column2=[Estado], 1, 64
Column3=[Hora], 1, 64

[DLG:IDD_PROPIEDADES_DIALOG]
Type=1
Class=CPropiedades1Dlg
ControlCount=36
Control1=IDOK,button,1342242817
Control2=IDCANCEL,button,1342242816
Control3=IDC_STATIC,static,1342308352
Control4=IDC_EDIT1,edit,1350631552
Control5=IDC_STATIC,static,1342308352
Control6=IDC_EDIT2,edit,1350631552
Control7=IDC_STATIC,static,1342308352
Control8=IDC_EDIT3,edit,1350631552
Control9=IDC_STATIC,static,1342308352
Control10=IDC_EDIT4,edit,1350631552
Control11=IDC_STATIC,static,1342308352
Control12=IDC_EDIT5,edit,1350631552
Control13=IDC_STATIC,static,1342308352
Control14=IDC_EDIT6,edit,1350631552
Control15=IDC_STATIC,static,1342308352
Control16=IDC_EDIT7,edit,1350631552
Control17=IDC_STATIC,static,1342308352
Control18=IDC_EDIT8,edit,1350631552
Control19=IDC_STATIC,static,1342308352
Control20=IDC_EDIT9,edit,1350631552
Control21=IDC_STATIC,static,1342308352
Control22=IDC_EDIT10,edit,1350631552
Control23=IDC_STATIC,static,1342308352
Control24=IDC_EDIT11,edit,1350631552
Control25=IDC_STATIC,static,1342308352
Control26=IDC_EDIT12,edit,1350631552
Control27=IDC_STATIC,static,1342308352
Control28=IDC_EDIT13,edit,1350631552
```

```
Control29=IDC_STATIC,static,1342308352
Control30=IDC_EDIT14,edit,1350631552
Control31=IDC_STATIC,button,1342177287
Control32=IDC_STATIC,button,1342177287
Control33=IDC_STATIC,button,1342177287
Control34=IDC_EDIT15,edit,1350631552
Control35=IDC_STATIC,static,1342308352
Control36=IDC_STATIC,button,1342177287

[CLS:CPropiedades1Dlg]
Type=0
HeaderFile=Propiedades1Dlg.h
ImplementationFile=Propiedades1Dlg.cpp
BaseClass=CDialog
Filter=D
LastObject=CPropiedades1Dlg
VirtualFilter=dWC

[DLG:IDD_CLIENTEDLG_DIALOG]
Type=1
Class=CClienteDlg
ControlCount=54
Control1=IDC_STATIC1,static,1342308352
Control2=IDC_STATIC,button,1342177287
Control3=IDC_WINSOCK1,{248DD896-BB45-11CF-9ABC-0080C7E7B78D},1342242816
Control4=IDC_STATIC,button,1342177287
Control5=IDC_STATIC,static,1342308352
Control6=IDC_EDIT01,edit,1350633600
Control7=IDC_STATIC,static,1342308352
Control8=IDC_EDIT02,edit,1350633600
Control9=IDC_STATIC,static,1342308352
Control10=IDC_EDIT03,edit,1350633600
Control11=IDC_STATIC,static,1342308352
Control12=IDC_EDIT04,edit,1350633600
Control13=IDC_STATIC,static,1342308352
Control14=IDC_EDIT05,edit,1350633600
Control15=IDC_STATIC,static,1342308352
Control16=IDC_EDIT06,edit,1350633600
Control17=IDC_STATIC,static,1342308352
Control18=IDC_EDIT07,edit,1350633600
Control19=IDC_STATIC,static,1342308352
Control20=IDC_EDIT08,edit,1350633600
Control21=IDC_STATIC,static,1342308352
Control22=IDC_EDIT09,edit,1350633600
Control23=IDC_STATIC,static,1342308352
Control24=IDC_EDIT010,edit,1350633600
Control25=IDC_STATIC,static,1342308352
Control26=IDC_EDIT011,edit,1350633600
Control27=IDC_STATIC,static,1342308352
Control28=IDC_EDIT012,edit,1350633600
Control29=IDC_STATIC,static,1342308352
Control30=IDC_EDIT013,edit,1350631552
Control31=IDC_STATIC,static,1342308352

```

```
<html>
<body>
<pre>
<h1>Build Log</h1>
<h3>
--------------------Configuration: ProyectoBD - Win32 Debug--------------------
</h3>
<h3>Command Lines</h3>
Creating command line "rc.exe /l 0xc0a /fo"Debug/ProyectoBD.res" /d "_DEBUG" "C:\Mis documentos\
Creating temporary file "C:\WINDOWS\TEMP\RSPC080.TMP" with contents
[
/nologo /MTd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /D "_MBCS" /Fp"Debug/Proye
"C:\Mis documentos\ProyectoBD\ProyectoBD\BaseDatosDlg.cpp"
"C:\Mis documentos\ProyectoBD\ProyectoBD\BaseDatosSet.cpp"
"C:\Mis documentos\ProyectoBD\ProyectoBD\ClienteDlg.cpp"
"C:\Mis documentos\ProyectoBD\ProyectoBD\mswinsockcontrol.cpp"
"C:\Mis documentos\ProyectoBD\ProyectoBD\Propiedades1Dlg.cpp"
"C:\Mis documentos\ProyectoBD\ProyectoBD\ProyectoBD.cpp"
"C:\Mis documentos\ProyectoBD\ProyectoBD\ProyectoBDDlg.cpp"
"C:\Mis documentos\ProyectoBD\ProyectoBD\PuertoParalelo.cpp"
]
Creating command line "cl.exe @C:\WINDOWS\TEMP\RSPC080.TMP"
Creating temporary file "C:\WINDOWS\TEMP\RSPC081.TMP" with contents
[
/nologo /MTd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /D "_MBCS" /Fp"Debug/Proye
"C:\Mis documentos\ProyectoBD\ProyectoBD\StdAfx.cpp"
]
Creating command line "cl.exe @C:\WINDOWS\TEMP\RSPC081.TMP"
Creating temporary file "C:\WINDOWS\TEMP\RSPC082.TMP" with contents
[
/nologo /subsystem:windows /incremental:yes /pdb:"Debug/ProyectoBD.pdb" /debug /machine:I386 /ou
".\Debug\BaseDatosDlg.obj"
".\Debug\BaseDatosSet.obj"
".\Debug\ClienteDlg.obj"
".\Debug\mswinsockcontrol.obj"
".\Debug\Propiedades1Dlg.obj"
".\Debug\ProyectoBD.obj"
".\Debug\ProyectoBDDlg.obj"
".\Debug\PuertoParalelo.obj"
".\Debug\StdAfx.obj"
".\Debug\ProyectoBD.res"
]
Creating command line "link.exe @C:\WINDOWS\TEMP\RSPC082.TMP"
<h3>Output Window</h3>
Compiling resources...
Compiling...
StdAfx.cpp
Compiling...
BaseDatosDlg.cpp
BaseDatosSet.cpp
ClienteDlg.cpp
mswinsockcontrol.cpp
Propiedades1Dlg.cpp
ProyectoBD.cpp
ProyectoBDDlg.cpp
PuertoParalelo.cpp
Generating Code...
Linking...



<h3>Results</h3>
ProyectoBD.exe - 0 error(s), 0 warning(s)
</pre>
```

```
</body>
</html>

```

```
1  //Microsoft Developer Studio generated resource script.
2  //
3  #include "resource.h"
4
5  #define APSTUDIO_READONLY_SYMBOLS
6  /////////////////////////////////////////////////////////////////////////////
7  //
8  // Generated from the TEXTINCLUDE 2 resource.
9  //
10 #include "afxres.h"
11
12 /////////////////////////////////////////////////////////////////////////////
13 #undef APSTUDIO_READONLY_SYMBOLS
14
15 /////////////////////////////////////////////////////////////////////////////
16 // Spanish (Modern) resources
17
18 #if !defined(AFX_RESOURCE_DLL) || defined(AFX_TARG_ESN)
19 #ifdef _WIN32
20 LANGUAGE LANG_SPANISH, SUBLANG_SPANISH_MODERN
21 #pragma code_page(1252)
22 #endif //_WIN32
23
24 #ifdef APSTUDIO_INVOKED
25 /////////////////////////////////////////////////////////////////////////////
26 //
27 // TEXTINCLUDE
28 //
29
30 1 TEXTINCLUDE DISCARDABLE
31 BEGIN
32     "resource.h\0"
33 END
34
35 2 TEXTINCLUDE DISCARDABLE
36 BEGIN
37     "#include ""afxres.h""\r\n"
38     "\0"
39 END
40
41 3 TEXTINCLUDE DISCARDABLE
42 BEGIN
43     "#define _AFX_NO_SPLITTER_RESOURCES\r\n"
44     "#define _AFX_NO_OLE_RESOURCES\r\n"
45     "#define _AFX_NO_TRACKER_RESOURCES\r\n"
46     "#define _AFX_NO_PROPERTY_RESOURCES\r\n"
47     "\r\n"
48     "#if !defined(AFX_RESOURCE_DLL) || defined(AFX_TARG_ESP)\r\n"
49     "#ifdef _WIN32\r\n"
50     "LANGUAGE 10, 3\r\n"
51     "#pragma code_page(1252)\r\n"
52     "#endif //_WIN32\r\n"
53     "#include ""res\\ProyectoBD.rc2""  // non-Microsoft Visual C++ edited resources\r\n"
54     "#include ""l.esp\\afxres.rc""          // Standard components\r\n"
55     "#endif\r\n"
56     "\0"
57 END
58
59 #endif    // APSTUDIO_INVOKED
60
61
62 /////////////////////////////////////////////////////////////////////////////
63 //
64 // Icon
```

```
65  //
66
67  // Icon with lowest ID value placed first to ensure application icon
68  // remains consistent on all systems.
69  IDR_MAINFRAME           ICON    DISCARDABLE     "res\\ProyectoBD.ico"
70
71  /////////////////////////////////////////////////////////////////////////////
72  //
73  // Dialog
74  //
75
76  IDD_ABOUTBOX DIALOG DISCARDABLE  0, 0, 235, 55
77  STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION | WS_SYSMENU
78  CAPTION "Acerca de ProyectoBD"
79  FONT 8, "MS Sans Serif"
80  BEGIN
81      ICON            IDR_MAINFRAME,IDC_STATIC,11,17,20,20
82      LTEXT           "ProyectoBD Versión 1.0",IDC_STATIC,40,10,119,8,
83                      SS_NOPREFIX
84      LTEXT           "Copyright (C) 2001",IDC_STATIC,40,25,119,8
85      DEFPUSHBUTTON   "Aceptar",IDOK,178,7,50,14,WS_GROUP
86  END
87
88  IDD_PROYECTOBD_DIALOG DIALOGEX 80, 65456, 389, 157
89  STYLE DS_MODALFRAME | WS_POPUP | WS_VISIBLE | WS_CAPTION | WS_SYSMENU
90  EXSTYLE WS_EX_APPWINDOW
91  CAPTION "ProyectoBD"
92  MENU IDR_MENU1
93  FONT 8, "MS Sans Serif"
94  BEGIN
95      LTEXT           "A HACER: Colocar aquí controles de cuadro de diálogo.",
96                      IDC_STATIC1,7,8,375,8
97      EDITTEXT        IDC_EDIT_NOMBRE,39,22,102,12,ES_AUTOHSCROLL
98      EDITTEXT        IDC_EDIT_ESTADO,39,34,102,12,ES_AUTOHSCROLL
99      EDITTEXT        IDC_EDIT_HORA,39,47,102,12,ES_AUTOHSCROLL
100     RTEXT           "Nombre:",IDC_STATIC,7,23,30,11
101     RTEXT           "Estado:",IDC_STATIC,7,36,30,11
102     RTEXT           "Hora:",IDC_STATIC,7,47,30,11
103     CONTROL         "Salida1",IDC_CHECK1,"Button",BS_AUTOCHECKBOX | BS_ICON |
104                     WS_TABSTOP,30,69,10,10
105     CONTROL         "Salida2",IDC_CHECK2,"Button",BS_AUTOCHECKBOX | BS_ICON |
106                     WS_TABSTOP,46,69,10,10
107     CONTROL         "Salida3",IDC_CHECK3,"Button",BS_AUTOCHECKBOX | BS_ICON |
108                     WS_TABSTOP,62,69,10,10
109     CONTROL         "Salida4",IDC_CHECK4,"Button",BS_AUTOCHECKBOX | BS_ICON |
110                     WS_TABSTOP,78,69,10,10
111     CONTROL         "Salida5",IDC_CHECK5,"Button",BS_AUTOCHECKBOX | BS_ICON |
112                     WS_TABSTOP,94,69,10,10
113     CONTROL         "Salida6",IDC_CHECK6,"Button",BS_AUTOCHECKBOX | BS_ICON |
114                     WS_TABSTOP,110,69,10,10
115     CONTROL         "Salida7",IDC_CHECK7,"Button",BS_AUTOCHECKBOX | BS_ICON |
116                     WS_TABSTOP,126,69,10,10
117     CONTROL         "Salida8",IDC_CHECK8,"Button",BS_AUTOCHECKBOX | BS_ICON |
118                     WS_TABSTOP,142,69,10,10
119     GROUPBOX        "Salidas",IDC_STATIC,15,60,150,23
120     CONTROL         "",IDC_WINSOCK1,"{248DD896-BB45-11CF-9ABC-0080C7E7B78D}",
121                     WS_TABSTOP,149,34,19,17
122     CONTROL         "Entrada1",IDC_CHECK9,"Button",BS_AUTOCHECKBOX | BS_ICON |
123                     WS_DISABLED | WS_TABSTOP,62,93,10,10
124     CONTROL         "Entrada2",IDC_CHECK10,"Button",BS_AUTOCHECKBOX |
125                     BS_ICON | WS_DISABLED | WS_TABSTOP,78,93,10,10
126     CONTROL         "Entrada3",IDC_CHECK11,"Button",BS_AUTOCHECKBOX |
127                     BS_ICON | WS_DISABLED | WS_TABSTOP,94,93,10,10
128     CONTROL         "Entrada4",IDC_CHECK12,"Button",BS_AUTOCHECKBOX |
```

```
                    BS_ICON | WS_DISABLED | WS_TABSTOP,110,93,10,10
    GROUPBOX        "Entradas",IDC_STATIC,15,85,150,23
    LTEXT           "1:",IDC_STATIC,179,35,10,10
    EDITTEXT        IDC_EDIT01,188,34,87,12,ES_AUTOHSCROLL | ES_READONLY
    LTEXT           "2:",IDC_STATIC,179,50,10,10
    EDITTEXT        IDC_EDIT02,188,47,87,12,ES_AUTOHSCROLL | ES_READONLY
    LTEXT           "3:",IDC_STATIC,179,63,10,10
    EDITTEXT        IDC_EDIT03,188,61,87,12,ES_AUTOHSCROLL | ES_READONLY
    LTEXT           "4:",IDC_STATIC,179,77,10,10
    EDITTEXT        IDC_EDIT04,188,75,87,12,ES_AUTOHSCROLL | ES_READONLY
    LTEXT           "5:",IDC_STATIC,179,91,10,10
    EDITTEXT        IDC_EDIT05,188,90,87,12,ES_AUTOHSCROLL | ES_READONLY
    LTEXT           "6:",IDC_STATIC,179,106,10,10
    EDITTEXT        IDC_EDIT06,188,103,87,12,ES_AUTOHSCROLL | ES_READONLY
    LTEXT           "7:",IDC_STATIC,179,119,10,10
    EDITTEXT        IDC_EDIT07,188,117,87,12,ES_AUTOHSCROLL | ES_READONLY
    LTEXT           "8:",IDC_STATIC,179,133,10,10
    EDITTEXT        IDC_EDIT08,188,131,87,12,ES_AUTOHSCROLL | ES_READONLY
    LTEXT           "1:",IDC_STATIC,279,36,10,10
    EDITTEXT        IDC_EDIT09,288,34,87,12,ES_AUTOHSCROLL | ES_READONLY
    LTEXT           "2:",IDC_STATIC,279,50,10,10
    EDITTEXT        IDC_EDIT010,288,48,87,12,ES_AUTOHSCROLL | ES_READONLY
    LTEXT           "3:",IDC_STATIC,279,64,10,10
    EDITTEXT        IDC_EDIT011,288,62,87,12,ES_AUTOHSCROLL | ES_READONLY
    LTEXT           "4:",IDC_STATIC,279,78,10,10
    EDITTEXT        IDC_EDIT012,288,76,87,12,ES_AUTOHSCROLL | ES_READONLY
    LTEXT           "IP:",IDC_STATIC,280,112,18,10
    EDITTEXT        IDC_EDIT013,302,109,73,12,ES_AUTOHSCROLL | ES_READONLY
    LTEXT           "Port:",IDC_STATIC,280,125,18,10
    EDITTEXT        IDC_EDIT014,302,123,73,12,ES_AUTOHSCROLL | ES_READONLY
    GROUPBOX        "Leyenda Salidas",IDC_STATIC,177,18,100,130
    GROUPBOX        "Leyenda Entradas",IDC_STATIC,277,18,100,75
    GROUPBOX        "Conexión",IDC_STATIC,278,97,100,51
    LTEXT           "Path:",IDC_STATIC,18,119,18,10
    EDITTEXT        IDC_EDIT16,18,129,145,12,ES_AUTOHSCROLL | ES_READONLY
    GROUPBOX        "Base de Datos",IDC_STATIC,15,110,150,38
END

IDD_BASEDATOSDLG_DIALOG DIALOG DISCARDABLE  0, 0, 260, 86
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION | WS_SYSMENU
CAPTION "Base de Datos"
FONT 8, "MS Sans Serif"
BEGIN
    LTEXT           "Static",IDC_STATIC1,7,7,133,9
    EDITTEXT        IDC_EDIT_NOMBRE,39,22,102,12,ES_AUTOHSCROLL
    EDITTEXT        IDC_EDIT_ESTADO,39,34,102,12,ES_AUTOHSCROLL
    EDITTEXT        IDC_EDIT_HORA,39,47,102,12,ES_AUTOHSCROLL
    PUSHBUTTON      "PRIMERO",IDC_BUTTON_PRIMERO,143,63,51,16
    PUSHBUTTON      "ANTERIOR",IDC_BUTTON_ANTERIOR,39,63,51,16
    PUSHBUTTON      "SIGUIENTE",IDC_BUTTON_SIGUIENTE,91,63,51,16
    PUSHBUTTON      "ULTIMO",IDC_BUTTON_ULTIMO,195,63,51,16
    PUSHBUTTON      "AÑADIR",IDC_BUTTON_INSERTAR,195,31,51,16
    PUSHBUTTON      "BORRAR",IDC_BUTTON_BORRAR,143,47,51,16
    PUSHBUTTON      "ACTUALIZAR",IDC_BUTTON_ACTUALIZAR,195,47,51,16
    PUSHBUTTON      "TODOS",IDC_BUTTON_BUSCARTODOS,143,31,51,16
    PUSHBUTTON      "LIMPIAR",IDC_BUTTON_LIMPIAR,143,15,51,16
    PUSHBUTTON      "BUSCAR",IDC_BUTTON_BUSCAR,195,15,51,16
    RTEXT           "Nombre:",IDC_STATIC,7,23,30,11
    RTEXT           "Estado:",IDC_STATIC,7,36,30,11
    RTEXT           "Hora:",IDC_STATIC,7,47,30,11
END

IDD_PROPIEDADES_DIALOG DIALOG DISCARDABLE  0, 0, 217, 188
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION | WS_SYSMENU
```

```
193 CAPTION "Propiedades"
194 FONT 8, "MS Sans Serif"
195 BEGIN
196     DEFPUSHBUTTON   "OK",IDOK,102,167,50,14
197     PUSHBUTTON      "Cancel",IDCANCEL,155,167,50,14
198     LTEXT           "1:",IDC_STATIC,7,22,10,10
199     EDITTEXT        IDC_EDIT1,16,20,87,12,ES_AUTOHSCROLL
200     LTEXT           "2:",IDC_STATIC,7,36,10,10
201     EDITTEXT        IDC_EDIT2,16,34,87,12,ES_AUTOHSCROLL
202     LTEXT           "3:",IDC_STATIC,7,50,10,10
203     EDITTEXT        IDC_EDIT3,16,48,87,12,ES_AUTOHSCROLL
204     LTEXT           "4:",IDC_STATIC,7,64,10,10
205     EDITTEXT        IDC_EDIT4,16,62,87,12,ES_AUTOHSCROLL
206     LTEXT           "5:",IDC_STATIC,7,78,10,10
207     EDITTEXT        IDC_EDIT5,16,76,87,12,ES_AUTOHSCROLL
208     LTEXT           "6:",IDC_STATIC,7,92,10,10
209     EDITTEXT        IDC_EDIT6,16,90,87,12,ES_AUTOHSCROLL
210     LTEXT           "7:",IDC_STATIC,7,106,10,10
211     EDITTEXT        IDC_EDIT7,16,104,87,12,ES_AUTOHSCROLL
212     LTEXT           "8:",IDC_STATIC,7,120,10,10
213     EDITTEXT        IDC_EDIT8,16,118,87,12,ES_AUTOHSCROLL
214     LTEXT           "1:",IDC_STATIC,107,23,10,10
215     EDITTEXT        IDC_EDIT9,116,21,87,12,ES_AUTOHSCROLL
216     LTEXT           "2:",IDC_STATIC,107,37,10,10
217     EDITTEXT        IDC_EDIT10,116,34,87,12,ES_AUTOHSCROLL
218     LTEXT           "3:",IDC_STATIC,107,50,10,10
219     EDITTEXT        IDC_EDIT11,116,49,87,12,ES_AUTOHSCROLL
220     LTEXT           "4:",IDC_STATIC,107,65,10,10
221     EDITTEXT        IDC_EDIT12,116,63,87,12,ES_AUTOHSCROLL
222     LTEXT           "IP:",IDC_STATIC,108,97,18,10
223     EDITTEXT        IDC_EDIT13,130,95,73,12,ES_AUTOHSCROLL
224     LTEXT           "Port:",IDC_STATIC,108,111,18,10
225     EDITTEXT        IDC_EDIT14,130,109,73,12,ES_AUTOHSCROLL
226     GROUPBOX        "Leyenda Salidas",IDC_STATIC,5,5,100,130
227     GROUPBOX        "Leyenda Entradas",IDC_STATIC,105,5,100,75
228     GROUPBOX        "Conexión",IDC_STATIC,106,80,100,55
229     EDITTEXT        IDC_EDIT15,28,146,175,12,ES_AUTOHSCROLL
230     LTEXT           "Path:",IDC_STATIC,9,148,18,10
231     GROUPBOX        "Base de Datos",IDC_STATIC,7,136,197,28
232 END
233
234 IDD_CLIENTEDLG_DIALOG DIALOG DISCARDABLE  0, 160, 389, 154
235 STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION | WS_SYSMENU
236 CAPTION "Dialog"
237 FONT 8, "MS Sans Serif"
238 BEGIN
239     LTEXT           "A HACER: Colocar aquí controles de cuadro de diálogo.",
240                     IDC_STATIC1,7,8,375,8
241     GROUPBOX        "Salidas",IDC_STATIC,15,97,150,25
242     CONTROL         "",IDC_WINSOCK1,"{248DD896-BB45-11CF-9ABC-0080C7E7B78D}",
243                     WS_TABSTOP,143,77,19,17
244     GROUPBOX        "Entradas",IDC_STATIC,15,124,150,23
245     LTEXT           "1:",IDC_STATIC,179,35,10,10
246     EDITTEXT        IDC_EDIT01,188,34,87,12,ES_AUTOHSCROLL | ES_READONLY
247     LTEXT           "2:",IDC_STATIC,179,50,10,10
248     EDITTEXT        IDC_EDIT02,188,47,87,12,ES_AUTOHSCROLL | ES_READONLY
249     LTEXT           "3:",IDC_STATIC,179,63,10,10
250     EDITTEXT        IDC_EDIT03,188,61,87,12,ES_AUTOHSCROLL | ES_READONLY
251     LTEXT           "4:",IDC_STATIC,179,77,10,10
252     EDITTEXT        IDC_EDIT04,188,75,87,12,ES_AUTOHSCROLL | ES_READONLY
253     LTEXT           "5:",IDC_STATIC,179,91,10,10
254     EDITTEXT        IDC_EDIT05,188,90,87,12,ES_AUTOHSCROLL | ES_READONLY
255     LTEXT           "6:",IDC_STATIC,179,106,10,10
256     EDITTEXT        IDC_EDIT06,188,103,87,12,ES_AUTOHSCROLL | ES_READONLY
```

```
257     LTEXT           "7:",IDC_STATIC,179,119,10,10
258     EDITTEXT        IDC_EDIT07,188,117,87,12,ES_AUTOHSCROLL | ES_READONLY
259     LTEXT           "8:",IDC_STATIC,179,133,10,10
260     EDITTEXT        IDC_EDIT08,188,131,87,12,ES_AUTOHSCROLL | ES_READONLY
261     LTEXT           "1:",IDC_STATIC,279,36,10,10
262     EDITTEXT        IDC_EDIT09,288,34,87,12,ES_AUTOHSCROLL | ES_READONLY
263     LTEXT           "2:",IDC_STATIC,279,50,10,10
264     EDITTEXT        IDC_EDIT010,288,48,87,12,ES_AUTOHSCROLL | ES_READONLY
265     LTEXT           "3:",IDC_STATIC,279,64,10,10
266     EDITTEXT        IDC_EDIT011,288,62,87,12,ES_AUTOHSCROLL | ES_READONLY
267     LTEXT           "4:",IDC_STATIC,279,78,10,10
268     EDITTEXT        IDC_EDIT012,288,76,87,12,ES_AUTOHSCROLL | ES_READONLY
269     LTEXT           "IP:",IDC_STATIC,280,112,18,10
270     EDITTEXT        IDC_EDIT013,302,109,73,12,ES_AUTOHSCROLL
271     LTEXT           "Port:",IDC_STATIC,280,125,18,10
272     EDITTEXT        IDC_EDIT014,302,123,73,12,ES_AUTOHSCROLL
273     GROUPBOX        "Leyenda Salidas",IDC_STATIC,177,18,100,130
274     GROUPBOX        "Leyenda Entradas",IDC_STATIC,277,18,100,75
275     LTEXT           "Recibido:",IDC_STATIC,18,29,31,10
276     EDITTEXT        IDC_EDIT16,18,39,145,12,ES_MULTILINE | ES_AUTOHSCROLL |
277                     ES_READONLY | WS_VSCROLL
278     GROUPBOX        "Socket",IDC_STATIC,15,20,150,76
279     PUSHBUTTON      "Conectar",IDC_BUTTON_CONECTAR,36,80,42,13
280     LTEXT           "Enviado:",IDC_STATIC,18,54,31,10
281     EDITTEXT        IDC_EDIT17,18,64,145,12,ES_MULTILINE | ES_AUTOHSCROLL |
282                     ES_READONLY | WS_VSCROLL
283     PUSHBUTTON      "Cerrar",IDC_BUTTON_CERRAR,100,80,42,13
284     GROUPBOX        "Conexión",IDC_STATIC,278,97,100,51
285     CONTROL         "Check1",IDC_CHK1,"Button",BS_AUTOCHECKBOX | BS_ICON |
286                     WS_TABSTOP,30,105,10,14
287     CONTROL         "Check1",IDC_CHK2,"Button",BS_AUTOCHECKBOX | BS_ICON |
288                     WS_TABSTOP,45,105,10,14
289     CONTROL         "Check1",IDC_CHK3,"Button",BS_AUTOCHECKBOX | BS_ICON |
290                     WS_TABSTOP,60,105,10,14
291     CONTROL         "Check1",IDC_CHK4,"Button",BS_AUTOCHECKBOX | BS_ICON |
292                     WS_TABSTOP,75,105,10,14
293     CONTROL         "Check1",IDC_CHK5,"Button",BS_AUTOCHECKBOX | BS_ICON |
294                     WS_TABSTOP,90,105,10,14
295     CONTROL         "Check1",IDC_CHK6,"Button",BS_AUTOCHECKBOX | BS_ICON |
296                     WS_TABSTOP,105,105,10,14
297     CONTROL         "Check1",IDC_CHK7,"Button",BS_AUTOCHECKBOX | BS_ICON |
298                     WS_TABSTOP,120,105,10,14
299     CONTROL         "Check1",IDC_CHK8,"Button",BS_AUTOCHECKBOX | BS_ICON |
300                     WS_TABSTOP,135,105,10,14
301     CONTROL         "Check1",IDC_CHK9,"Button",BS_AUTOCHECKBOX | BS_ICON |
302                     WS_DISABLED | WS_TABSTOP,57,130,10,14
303     CONTROL         "Check1",IDC_CHK10,"Button",BS_AUTOCHECKBOX | BS_ICON |
304                     WS_DISABLED | WS_TABSTOP,73,130,10,14
305     CONTROL         "Check1",IDC_CHK11,"Button",BS_AUTOCHECKBOX | BS_ICON |
306                     WS_DISABLED | WS_TABSTOP,89,130,10,14
307     CONTROL         "Check1",IDC_CHK12,"Button",BS_AUTOCHECKBOX | BS_ICON |
308                     WS_DISABLED | WS_TABSTOP,105,130,10,14
309 END
310
311
312 #ifndef _MAC
313 /////////////////////////////////////////////////////////////////////////////
314 //
315 // Version
316 //
317
318 VS_VERSION_INFO VERSIONINFO
319  FILEVERSION 1,0,0,1
320  PRODUCTVERSION 1,0,0,1
```

```
 FILEFLAGSMASK 0x3fL
#ifdef _DEBUG
 FILEFLAGS 0x1L
#else
 FILEFLAGS 0x0L
#endif
 FILEOS 0x4L
 FILETYPE 0x1L
 FILESUBTYPE 0x0L
BEGIN
    BLOCK "StringFileInfo"
    BEGIN
        BLOCK "0C0A04B0"
        BEGIN
            VALUE "CompanyName", "\0"
            VALUE "FileDescription", "Aplicación MFC ProyectoBD\0"
            VALUE "FileVersion", "1, 0, 0, 1\0"
            VALUE "InternalName", "ProyectoBD\0"
            VALUE "LegalCopyright", "Copyright (C) 2001\0"
            VALUE "LegalTrademarks", "\0"
            VALUE "OriginalFilename", "ProyectoBD.EXE\0"
            VALUE "ProductName", "Aplicación ProyectoBD\0"
            VALUE "ProductVersion", "1, 0, 0, 1\0"
        END
    END
    BLOCK "VarFileInfo"
    BEGIN
        VALUE "Translation", 0xc0a, 1200
    END
END

#endif    // !_MAC


/////////////////////////////////////////////////////////////////////////////
//
// DESIGNINFO
//

#ifdef APSTUDIO_INVOKED
GUIDELINES DESIGNINFO DISCARDABLE
BEGIN
    IDD_ABOUTBOX, DIALOG
    BEGIN
        LEFTMARGIN, 7
        RIGHTMARGIN, 228
        TOPMARGIN, 7
        BOTTOMMARGIN, 48
    END

    IDD_PROYECTOBD_DIALOG, DIALOG
    BEGIN
        LEFTMARGIN, 7
        RIGHTMARGIN, 382
        TOPMARGIN, 7
        BOTTOMMARGIN, 150
    END

    IDD_BASEDATOSDLG_DIALOG, DIALOG
    BEGIN
        LEFTMARGIN, 7
        RIGHTMARGIN, 253
        TOPMARGIN, 7
        BOTTOMMARGIN, 79
```

```
385     END
386
387     IDD_PROPIEDADES_DIALOG, DIALOG
388     BEGIN
389         LEFTMARGIN, 7
390         RIGHTMARGIN, 204
391         TOPMARGIN, 7
392         BOTTOMMARGIN, 181
393     END
394
395     IDD_CLIENTEDLG_DIALOG, DIALOG
396     BEGIN
397         LEFTMARGIN, 7
398         RIGHTMARGIN, 382
399         TOPMARGIN, 7
400         BOTTOMMARGIN, 147
401     END
402 END
403 #endif    // APSTUDIO_INVOKED
404
405
406 /////////////////////////////////////////////////////////////////////////////
407 //
408 // Menu
409 //
410
411 IDR_MENU1 MENU DISCARDABLE
412 BEGIN
413     POPUP "Acción"
414     BEGIN
415         MENUITEM "Base de Datos",               ID_ACCIN_BASEDEDATOS
416         MENUITEM "Propiedades",                 ID_ACCIN_PROPIEDADES
417         MENUITEM "Cliente",                     ID_ACCIN_CLIENTE
418     END
419     POPUP "Salidas"
420     BEGIN
421         MENUITEM "Salida1",                     ID_SALIDAS_SALIDA1
422         MENUITEM "Salida2",                     ID_SALIDAS_SALIDA2
423         MENUITEM "Salida3",                     ID_SALIDAS_SALIDA3
424         MENUITEM "Salida4",                     ID_SALIDAS_SALIDA4
425         MENUITEM "Salida5",                     ID_SALIDAS_SALIDA5
426         MENUITEM "Salida6",                     ID_SALIDAS_SALIDA6
427         MENUITEM "Salida7",                     ID_SALIDAS_SALIDA7
428         MENUITEM "Salida8",                     ID_SALIDAS_SALIDA8
429     END
430 END
431
432
433 /////////////////////////////////////////////////////////////////////////////
434 //
435 // Dialog Info
436 //
437
438 IDD_PROYECTOBD_DIALOG DLGINIT
439 BEGIN
440     IDC_WINSOCK1, 0x376, 120, 0
441 0x0024, 0x0000, 0x0032, 0x0063, 0x0034, 0x0039, 0x0066, 0x0038, 0x0030,
442 0x0030, 0x002d, 0x0063, 0x0032, 0x0064, 0x0064, 0x002d, 0x0031, 0x0031,
443 0x0063, 0x0066, 0x002d, 0x0039, 0x0061, 0x0064, 0x0036, 0x002d, 0x0030,
444 0x0030, 0x0038, 0x0030, 0x0063, 0x0037, 0x0065, 0x0037, 0x0062, 0x0037,
445 0x0038, 0x0064, 0x4321, 0x1234, 0x0008, 0x0000, 0x02e5, 0x0000, 0x02e5,
446 0x0000, 0xd892, 0x248d, 0x0000, 0x0006, 0x0000, 0x0000, 0x0000, 0x0000,
447 0x0000, 0x0000, 0x0000, 0x0000, 0x5ad8, 0x0054,
448     0
```

```
449 END
450
451 IDD_CLIENTEDLG_DIALOG DLGINIT
452 BEGIN
453     IDC_WINSOCK1, 0x376, 116, 0
454 0x0024, 0x0000, 0x0032, 0x0063, 0x0034, 0x0039, 0x0066, 0x0038, 0x0030,
455 0x0030, 0x002d, 0x0063, 0x0032, 0x0064, 0x0064, 0x002d, 0x0031, 0x0031,
456 0x0063, 0x0066, 0x002d, 0x0039, 0x0061, 0x0064, 0x0036, 0x002d, 0x0030,
457 0x0030, 0x0038, 0x0030, 0x0063, 0x0037, 0x0065, 0x0037, 0x0062, 0x0037,
458 0x0038, 0x0064, 0x4321, 0x1234, 0x0008, 0x0000, 0x02e5, 0x0000, 0x02e5,
459 0x0000, 0xd892, 0x248d, 0x0000, 0x0006, 0x0000, 0x0000, 0x0000, 0x0000,
460 0x0000, 0x0000, 0x0000, 0x0000,
461     0
462 END
463
464
465 /////////////////////////////////////////////////////////////////////////////
466 //
467 // String Table
468 //
469
470 STRINGTABLE DISCARDABLE
471 BEGIN
472     IDS_ABOUTBOX            "&Acerca de ProyectoBD..."
473 END
474
475 #endif    // Spanish (Modern) resources
476 /////////////////////////////////////////////////////////////////////////////
477
478
479
480 #ifndef APSTUDIO_INVOKED
481 /////////////////////////////////////////////////////////////////////////////
482 //
483 // Generated from the TEXTINCLUDE 3 resource.
484 //
485 #define _AFX_NO_SPLITTER_RESOURCES
486 #define _AFX_NO_OLE_RESOURCES
487 #define _AFX_NO_TRACKER_RESOURCES
488 #define _AFX_NO_PROPERTY_RESOURCES
489
490 #if !defined(AFX_RESOURCE_DLL) || defined(AFX_TARG_ESP)
491 #ifdef _WIN32
492 LANGUAGE 10, 3
493 #pragma code_page(1252)
494 #endif //_WIN32
495 #include "res\ProyectoBD.rc2"  // non-Microsoft Visual C++ edited resources
496 #include "l.esp\afxres.rc"          // Standard components
497 #endif
498
499 /////////////////////////////////////////////////////////////////////////////
500 #endif    // not APSTUDIO_INVOKED
501
502
```

```cpp
// PuertoParalelo.cpp: implementation of the CPuertoParalelo class.
//
//////////////////////////////////////////////////////////////////////

#include "stdafx.h"
#include "PuertoParalelo.h"

#ifdef _DEBUG
#undef THIS_FILE
static char THIS_FILE[]=__FILE__;
#define new DEBUG_NEW
#endif

//////////////////////////////////////////////////////////////////////
// Construction/Destruction
//////////////////////////////////////////////////////////////////////

CPuertoParalelo::CPuertoParalelo()
{

}

CPuertoParalelo::~CPuertoParalelo()
{

}


int CPuertoParalelo::Leer()
{
    return _inp(0x379);
}

void CPuertoParalelo::Escribir(int dato)
{
    _outp(0x378,dato);
}

void CPuertoParalelo::Salida1(bool estado)
{
    if (estado) _outp(0x378,_inp(0x378)|0x01);
    else _outp(0x378,_inp(0x378)&(~0x01));
}

void CPuertoParalelo::Salida2(bool estado)
{
    if (estado) _outp(0x378,_inp(0x378)|0x02);
    else _outp(0x378,_inp(0x378)&(~0x02));
}

void CPuertoParalelo::Salida3(bool estado)
{
    if (estado) _outp(0x378,_inp(0x378)|0x04);
    else _outp(0x378,_inp(0x378)&(~0x04));
}

void CPuertoParalelo::Salida4(bool estado)
{
    if (estado) _outp(0x378,_inp(0x378)|0x08);
    else _outp(0x378,_inp(0x378)&(~0x08));
}

void CPuertoParalelo::Salida5(bool estado)
{
```

```
65      if (estado) _outp(0x378,_inp(0x378)|0x10);
66      else _outp(0x378,_inp(0x378)&(~0x10));
67  }
68
69  void CPuertoParalelo::Salida6(bool estado)
70  {
71      if (estado) _outp(0x378,_inp(0x378)|0x20);
72      else _outp(0x378,_inp(0x378)&(~0x20));
73  }
74
75  void CPuertoParalelo::Salida7(bool estado)
76  {
77      if (estado) _outp(0x378,_inp(0x378)|0x40);
78      else _outp(0x378,_inp(0x378)&(~0x40));
79  }
80
81  void CPuertoParalelo::Salida8(bool estado)
82  {
83      if (estado) _outp(0x378,_inp(0x378)|0x80);
84      else _outp(0x378,_inp(0x378)&(~0x80));
85  }
86
87  int CPuertoParalelo::Entrada1()
88  {
89      return ((_inp(0x379)&0x08)>>3);
90  }
91  int CPuertoParalelo::Entrada2()
92  {
93      return ((_inp(0x379)&0x10)>>4);
94  }
95  int CPuertoParalelo::Entrada3()
96  {
97      return ((_inp(0x379)&0x20)>>5);
98  }
99  int CPuertoParalelo::Entrada4()
100 {
101     return ((_inp(0x379)&0x40)>>6);
102 }
103
```

```
1  // PuertoParalelo.h: interface for the CPuertoParalelo class.
2  //
3  //////////////////////////////////////////////////////////////////////
4
5  #if !defined(AFX_PUERTOPARALELO_H__4E5D8741_A1E6_11D5_97E1_A19932F01649__INCLUDED_)
6  #define AFX_PUERTOPARALELO_H__4E5D8741_A1E6_11D5_97E1_A19932F01649__INCLUDED_
7
8  #if _MSC_VER > 1000
9  #pragma once
10 #endif // _MSC_VER > 1000
11
12 #include <conio.h>  //Libreria donde se encuentran las funciones
13                     // _inp() y _outp()
14
15 class CPuertoParalelo
16 {
17 public:
18     CPuertoParalelo();
19     virtual ~CPuertoParalelo();
20
21     int Entrada1();
22     int Entrada2();
23     int Entrada3();
24     int Entrada4();
25     void Salida1(bool estado);
26     void Salida2(bool estado);
27     void Salida3(bool estado);
28     void Salida4(bool estado);
29     void Salida5(bool estado);
30     void Salida6(bool estado);
31     void Salida7(bool estado);
32     void Salida8(bool estado);
33     void Escribir(int dato);
34     int Leer();
35 };
36
37 #endif // !defined(AFX_PUERTOPARALELO_H__4E5D8741_A1E6_11D5_97E1_A19932F01649__INCLUDED_)
38
```

```
========================================================================
       MICROSOFT FOUNDATION CLASS LIBRARY : ProyectoBD
========================================================================


AppWizard has created this ProyectoBD application for you.  This application
not only demonstrates the basics of using the Microsoft Foundation classes
but is also a starting point for writing your application.

This file contains a summary of what you will find in each of the files that
make up your ProyectoBD application.

ProyectoBD.dsp
    This file (the project file) contains information at the project level and
    is used to build a single project or subproject. Other users can share the
    project (.dsp) file, but they should export the makefiles locally.

ProyectoBD.h
    This is the main header file for the application.  It includes other
    project specific headers (including Resource.h) and declares the
    CProyectoBDApp application class.

ProyectoBD.cpp
    This is the main application source file that contains the application
    class CProyectoBDApp.

ProyectoBD.rc
    This is a listing of all of the Microsoft Windows resources that the
    program uses.  It includes the icons, bitmaps, and cursors that are stored
    in the RES subdirectory.  This file can be directly edited in Microsoft
        Visual C++.

ProyectoBD.clw
    This file contains information used by ClassWizard to edit existing
    classes or add new classes.  ClassWizard also uses this file to store
    information needed to create and edit message maps and dialog data
    maps and to create prototype member functions.

res\ProyectoBD.ico
    This is an icon file, which is used as the application's icon.  This
    icon is included by the main resource file ProyectoBD.rc.

res\ProyectoBD.rc2
    This file contains resources that are not edited by Microsoft
        Visual C++.  You should place all resources not editable by
        the resource editor in this file.




/////////////////////////////////////////////////////////////////////////////

AppWizard creates one dialog class:

ProyectoBDDlg.h, ProyectoBDDlg.cpp - the dialog
    These files contain your CProyectoBDDlg class.  This class defines
    the behavior of your application's main dialog.  The dialog's
    template is in ProyectoBD.rc, which can be edited in Microsoft
        Visual C++.


/////////////////////////////////////////////////////////////////////////////
Other standard files:

```

```
StdAfx.h, StdAfx.cpp
    These files are used to build a precompiled header (PCH) file
    named ProyectoBD.pch and a precompiled types file named StdAfx.obj.

Resource.h
    This is the standard header file, which defines new resource IDs.
    Microsoft Visual C++ reads and updates this file.

/////////////////////////////////////////////////////////////////////////////
Other notes:

AppWizard uses "TODO:" to indicate parts of the source code you
should add to or customize.

If your application uses MFC in a shared DLL, and your application is
in a language other than the operating system's current language, you
will need to copy the corresponding localized resources MFC42XXX.DLL
from the Microsoft Visual C++ CD-ROM onto the system or system32 directory,
and rename it to be MFCLOC.DLL.  ("XXX" stands for the language abbreviation.
For example, MFC42DEU.DLL contains resources translated to German.)  If you
don't do this, some of the UI elements of your application will remain in the
language of the operating system.

/////////////////////////////////////////////////////////////////////////////

```

```
1  //{{NO_DEPENDENCIES}}
2  // Microsoft Developer Studio generated include file.
3  // Used by ProyectoBD.rc
4  //
5  #define IDM_ABOUTBOX                    0x0010
6  #define IDD_ABOUTBOX                    100
7  #define IDS_ABOUTBOX                    101
8  #define IDD_PROYECTOBD_DIALOG           102
9  #define IDD_BASEDATOSDLG_DIALOG         103
10 #define IDD_CLIENTEDLG_DIALOG           105
11 #define IDR_MAINFRAME                   128
12 #define IDR_MENU1                       129
13 #define IDD_PROPIEDADES_DIALOG          132
14 #define IDC_EDIT_NOMBRE                 1000
15 #define IDC_EDIT_ESTADO                 1001
16 #define IDC_EDIT_HORA                   1002
17 #define IDC_BUTTON_PRIMERO              1003
18 #define IDC_BUTTON_ANTERIOR             1004
19 #define IDC_BUTTON_SIGUIENTE            1005
20 #define IDC_BUTTON_ULTIMO               1006
21 #define IDC_BUTTON_INSERTAR             1007
22 #define IDC_BUTTON_BORRAR               1008
23 #define IDC_BUTTON_ACTUALIZAR           1009
24 #define IDC_BUTTON_BUSCARTODOS          1010
25 #define IDC_BUTTON_LIMPIAR              1011
26 #define IDC_BUTTON_BUSCAR               1012
27 #define IDC_STATIC1                     1013
28 #define IDC_CHECK1                      1015
29 #define IDC_CHECK2                      1016
30 #define IDC_CHECK3                      1017
31 #define IDC_CHECK4                      1018
32 #define IDC_CHECK5                      1019
33 #define IDC_CHECK6                      1020
34 #define IDC_CHECK7                      1021
35 #define IDC_CHECK8                      1022
36 #define IDC_CHECK9                      1023
37 #define IDC_WINSOCK1                    1024
38 #define IDC_CHECK10                     1025
39 #define IDC_EDIT1                       1025
40 #define IDC_CHECK11                     1026
41 #define IDC_EDIT2                       1026
42 #define IDC_CHECK12                     1027
43 #define IDC_EDIT3                       1027
44 #define IDC_EDIT4                       1028
45 #define IDC_EDIT5                       1029
46 #define IDC_EDIT6                       1030
47 #define IDC_EDIT7                       1031
48 #define IDC_EDIT8                       1032
49 #define IDC_EDIT9                       1033
50 #define IDC_EDIT10                      1034
51 #define IDC_EDIT11                      1035
52 #define IDC_EDIT12                      1036
53 #define IDC_EDIT13                      1037
54 #define IDC_EDIT14                      1038
55 #define IDC_EDIT05                      1039
56 #define IDC_EDIT15                      1039
57 #define IDC_EDIT06                      1040
58 #define IDC_EDIT07                      1041
59 #define IDC_EDIT01                      1042
60 #define IDC_EDIT02                      1043
61 #define IDC_EDIT03                      1044
62 #define IDC_EDIT08                      1045
63 #define IDC_EDIT09                      1046
64 #define IDC_EDIT010                     1047
```

```
65  #define IDC_EDIT011                     1048
66  #define IDC_EDIT012                     1049
67  #define IDC_EDIT013                     1050
68  #define IDC_EDIT014                     1051
69  #define IDC_EDIT04                      1052
70  #define IDC_EDIT16                      1053
71  #define IDC_BUTTON_CONECTAR             1054
72  #define IDC_EDIT17                      1055
73  #define IDC_BUTTON_CERRAR               1056
74  #define IDC_CHK1                        1057
75  #define IDC_CHK2                        1058
76  #define IDC_CHK3                        1059
77  #define IDC_CHK4                        1060
78  #define IDC_CHK5                        1061
79  #define IDC_CHK6                        1062
80  #define IDC_CHK7                        1063
81  #define IDC_CHK8                        1064
82  #define IDC_CHK9                        1065
83  #define IDC_CHK10                       1066
84  #define IDC_CHK11                       1067
85  #define IDC_CHK12                       1068
86  #define ID_ACCIN_BASEDEDATOS            32771
87  #define ID_SALIDAS_SALIDA1              32772
88  #define ID_SALIDAS_SALIDA2              32773
89  #define ID_SALIDAS_SALIDA3              32774
90  #define ID_SALIDAS_SALIDA4              32775
91  #define ID_SALIDAS_SALIDA5              32776
92  #define ID_SALIDAS_SALIDA6              32777
93  #define ID_SALIDAS_SALIDA7              32778
94  #define ID_SALIDAS_SALIDA8              32779
95  #define ID_ACCIN_PROPIEDADES            32780
96  #define ID_ACCIN_CLIENTE                32781
97
98  // Next default values for new objects
99  //
100 #ifdef APSTUDIO_INVOKED
101 #ifndef APSTUDIO_READONLY_SYMBOLS
102 #define _APS_NEXT_RESOURCE_VALUE        135
103 #define _APS_NEXT_COMMAND_VALUE         32782
104 #define _APS_NEXT_CONTROL_VALUE         1069
105 #define _APS_NEXT_SYMED_VALUE           106
106 #endif
107 #endif
108
```

```
1  // stdafx.cpp : source file that includes just the standard includes
2  //  ProyectoBD.pch will be the pre-compiled header
3  //  stdafx.obj will contain the pre-compiled type information
4
5  #include "stdafx.h"
6
7
8
9
```

```
1  // stdafx.h : include file for standard system include files,
2  //  or project specific include files that are used frequently, but
3  //      are changed infrequently
4  //
5
6  #if !defined(AFX_STDAFX_H__9210F9F8_A9F7_11D5_97E1_FCCD2DB3034F__INCLUDED_)
7  #define AFX_STDAFX_H__9210F9F8_A9F7_11D5_97E1_FCCD2DB3034F__INCLUDED_
8
9  #if _MSC_VER > 1000
10 #pragma once
11 #endif // _MSC_VER > 1000
12
13 #define VC_EXTRALEAN        // Exclude rarely-used stuff from Windows headers
14
15 #include <afxwin.h>         // MFC core and standard components
16 #include <afxext.h>         // MFC extensions
17 #include <afxdisp.h>        // MFC Automation classes
18 #include <afxdtctl.h>       // MFC support for Internet Explorer 4 Common Controls
19 #ifndef _AFX_NO_AFXCMN_SUPPORT
20 #include <afxcmn.h>         // MFC support for Windows Common Controls
21 #endif // _AFX_NO_AFXCMN_SUPPORT
22
23 #include <afxdb.h>
24 #include <afxdao.h>
25
26 //{{AFX_INSERT_LOCATION}}
27 // Microsoft Visual C++ will insert additional declarations immediately before the previous line
28
29 #endif // !defined(AFX_STDAFX_H__9210F9F8_A9F7_11D5_97E1_FCCD2DB3034F__INCLUDED_)
30
```

Proyecto Final de Carrera:

CONTROL REMOTO MEDIANTE COMUNICACIÓN TCP/IP DE RADIOAYUDAS AERONAUTICAS

5.PRESENTACION

- *Ingeniero Superior en Automática y Electrónica Industrial* — Agustín Baubí Rovira
- *Profesor Ponente* — Ernest Gil Dolcet

Febrero 2002

Proyecto Final de Carrera:

CONTROL REMOTO MEDIANTE COMUNICACIÓN TCP/IP DE RADIOAYUDAS AERONAUTICAS.

•*Ingeniero Superior en Automática y Electrónica Industrial* Agustí Baubí Rovira

•*Profesor Ponente* Ernest Gil Dolcet

Febrero 2002

Objetivos del proyecto

- Control y monitorización remota de equipos electrónicos aeronáuticos.
  - Controlar los equipos localmente mediante el puerto paralelo del PC.
  - Controlar los equipos remotamente mediante comunicación remota cliente-servidor (Ethernet).
  - Registro de eventos en una base de datos local.

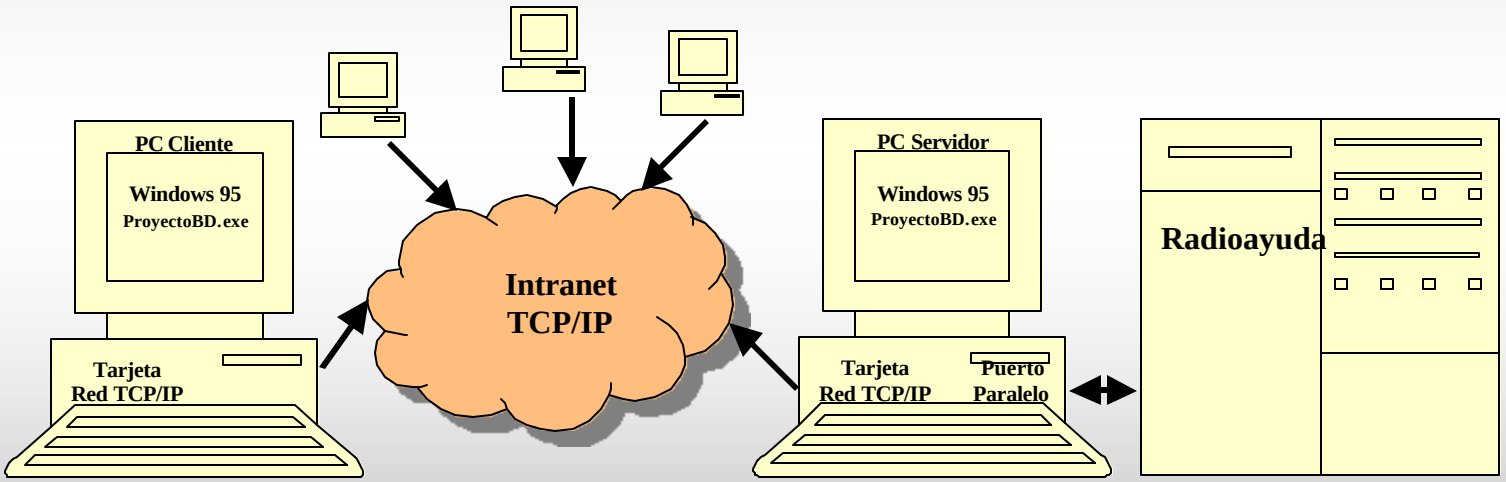
Radioayudas Aeronáuticas

- Equipos electrónicos de radiofrecuencia que proporcionan ayuda a la navegación aérea a las aeronaves.
- Las principales señales generadas son auto comprobadas por unos equipos monitores internos.
- Señales de los equipos que se deben incluir en un control remoto.

Entradas	Salidas
Entrada1: Equipo ON/OFF	Salida1: Equipo Principal ON en Antena
Entrada2: Equipo en antena Principal / secundario	Salida2: Equipo Principal ON en Carga
Entrada3: Monitores Bypass	Salida3: Equipo Principal OFF
Entrada4: Alarma de Mantenimiento	Salida4: Equipo Secundario ON en Antena
	Salida5: Equipo Secundario ON en Carga
	Salida6: Equipo Secundario OFF
	Salida7: Monitores Bypass
	Salida8: Reset alarma de Mantenimiento

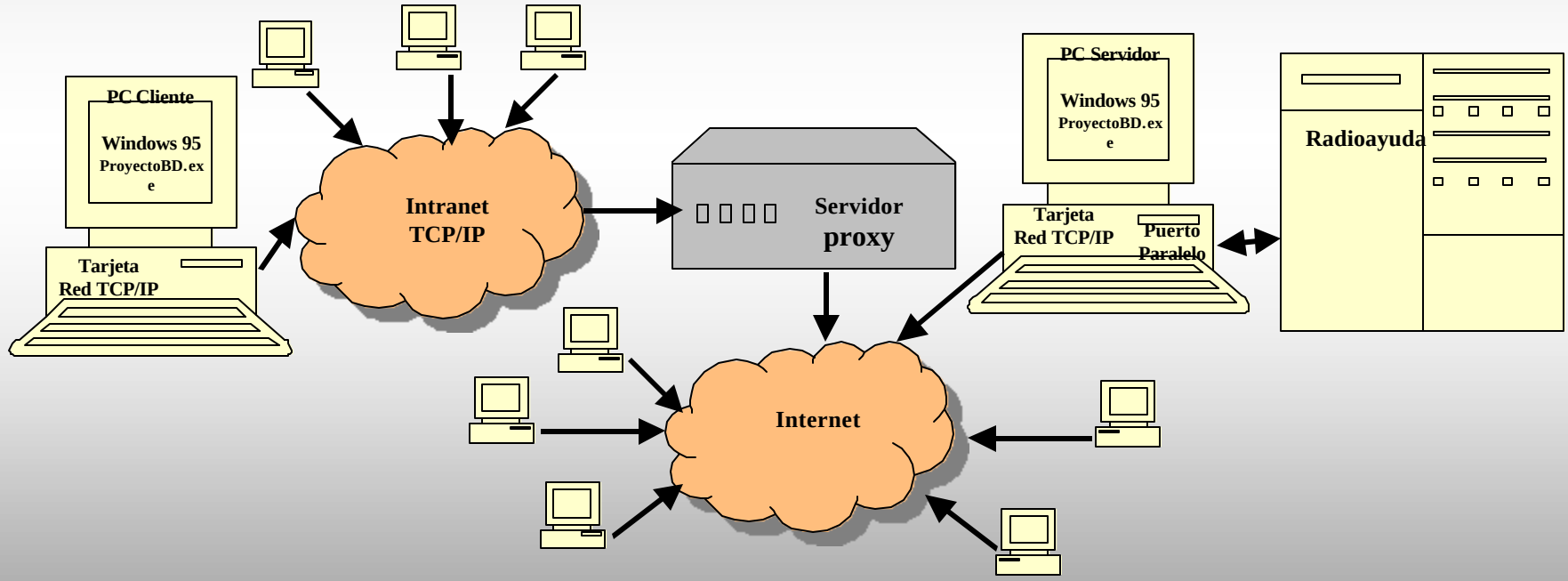
Configuración del Hardware

- **Intranet Local**



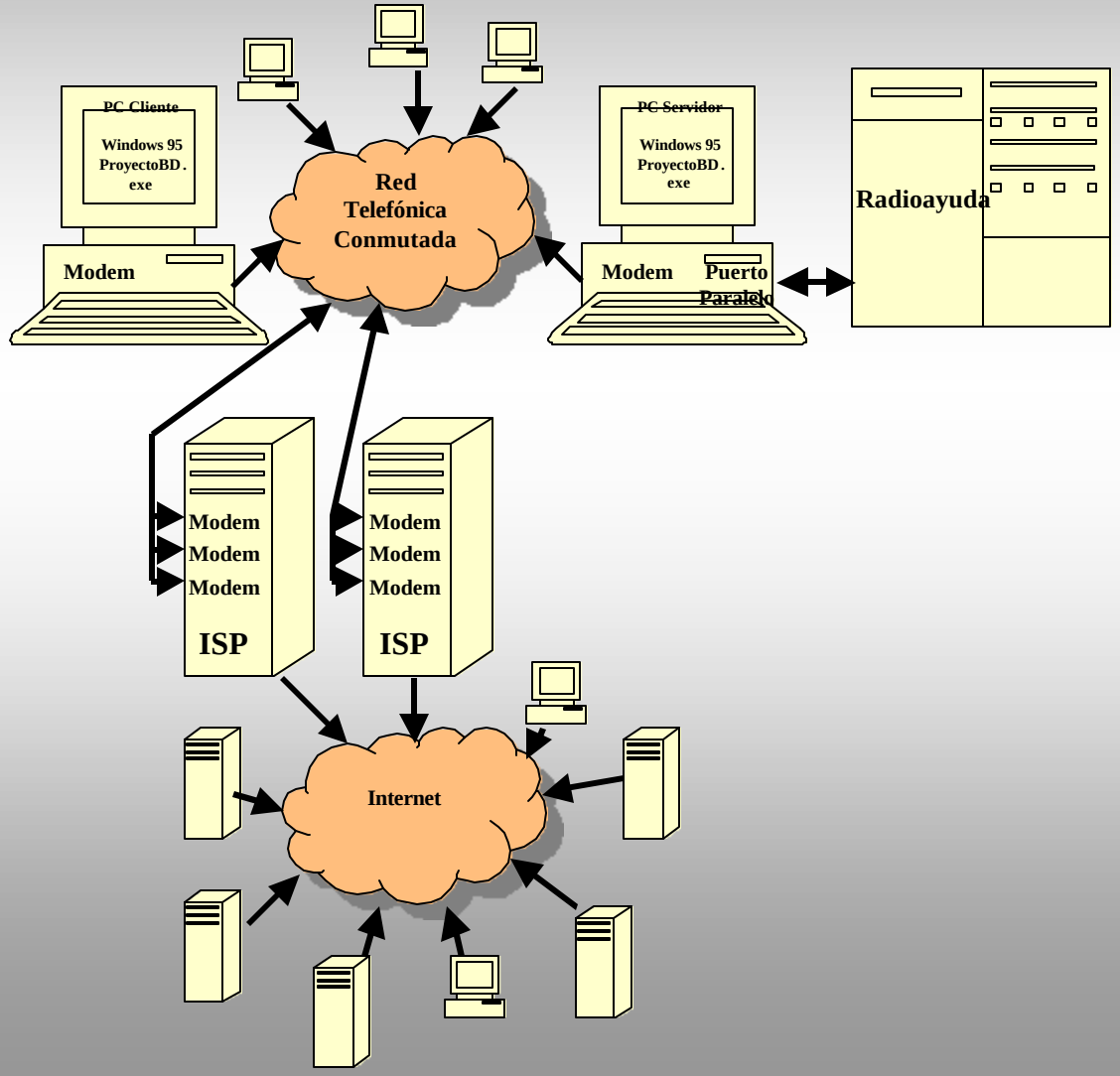
Configuración del Hardware

- **Intranet con conexión a Internet a través de un proxy**

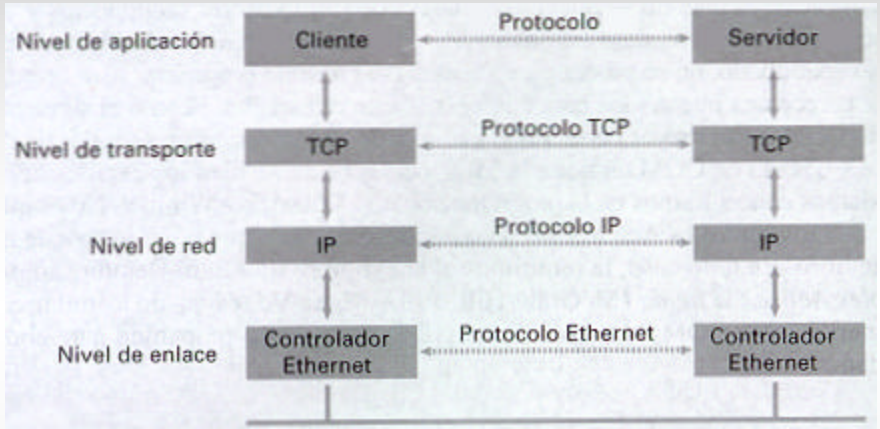


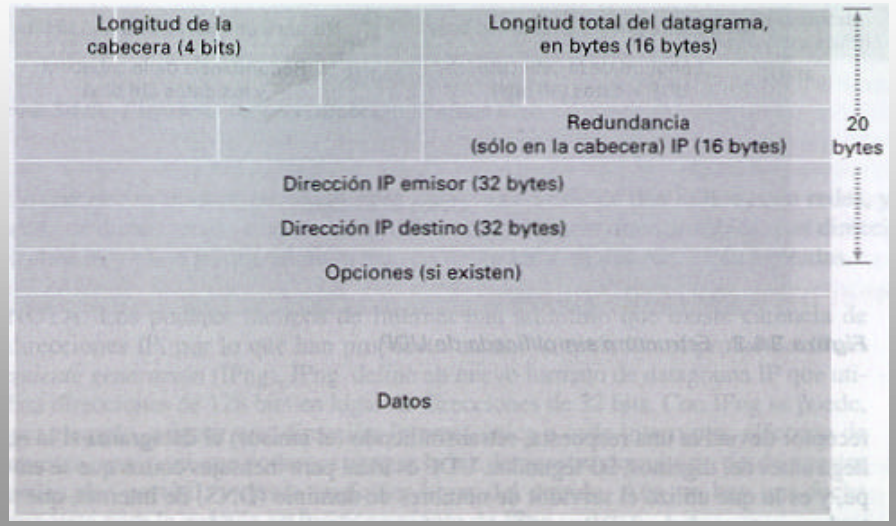
Configuración del Hardware

- **Conexión a Internet a través de ISP**



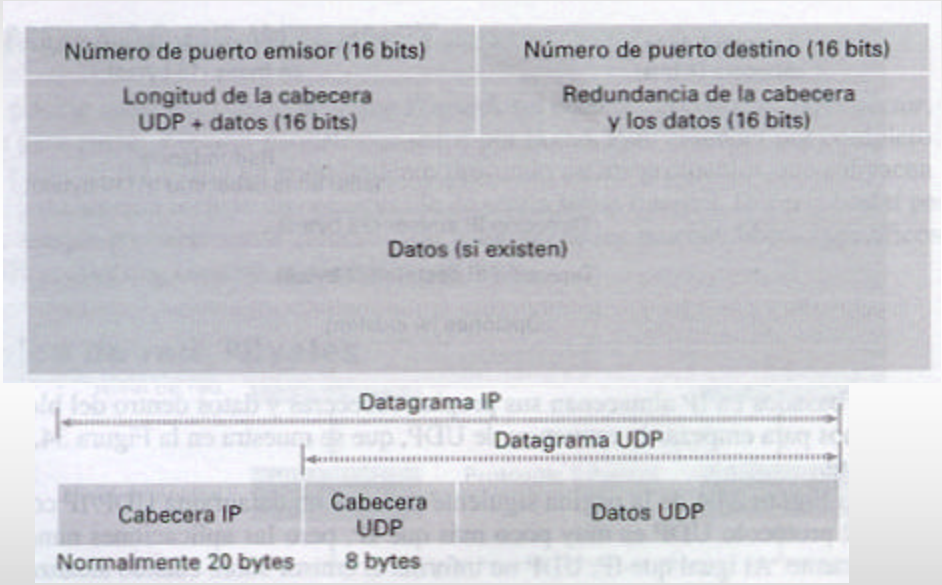
TCP/IP




Longitud de la cabecera (4 bits)
Longitud total del datagrama, en bytes (16 bytes)
Redundancia (sólo en la cabecera) IP (16 bytes)
20 bytes
Dirección IP emisor (32 bytes)
Dirección IP destino (32 bytes)
Opciones (si existen)
Datos

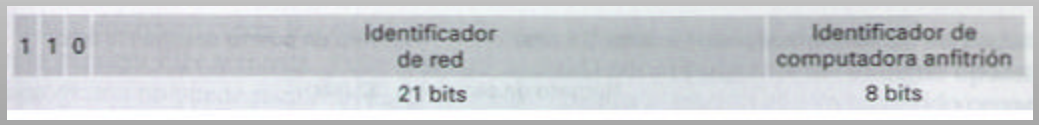
- Redes TCP/IP
  - Internet
  - Intranet
  - Extranet
- Protocolos de red: Niveles
- Comunicación cliente-servidor
- El Protocolo Internet (IP)
  - Datagramas
  - Direcciones de 32 bits

TCP/IP

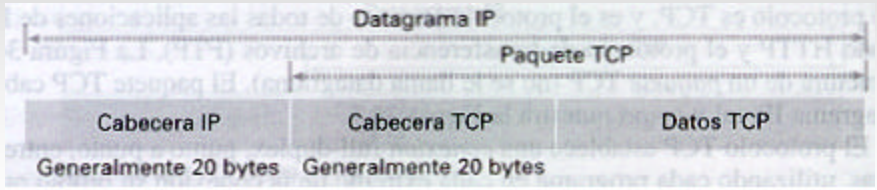

Número de puerto emisor (16 bits)
Número de puerto destino (16 bits)
Longitud de la cabecera UDP + datos (16 bits)
Redundancia de la cabecera y los datos (16 bits)
Datos (si existen)
Datagrama IP
Datagrama UDP
Cabecera IP
Cabecera UDP
Datos UDP
Normalmente 20 bytes
8 bytes

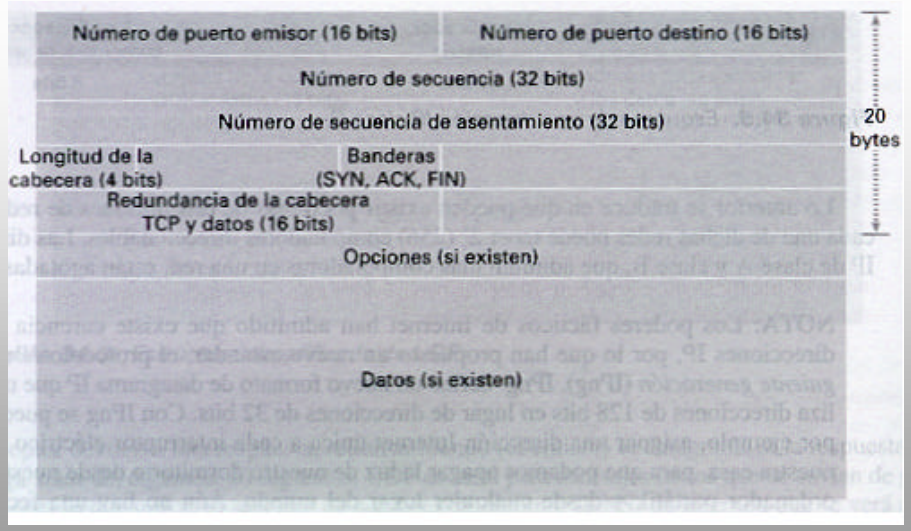
- Protocolo de datagramas de usuario (UDP/IP)
  - Puertos de 16 bits
  - No informa al emisor sobre cuándo alcanza el datagrama su destino.

- Formato de direcciones IP: 2^{32} máquinas con @ unívoca



TCP/IP




Número de puerto emisor (16 bits)
Número de puerto destino (16 bits)
Número de secuencia (32 bits)
Número de secuencia de asentamiento (32 bits)
Longitud de la cabecera (4 bits)
Banderas (SYN, ACK, FIN)
Redundancia de la cabecera TCP y datos (16 bits)
Opciones (si existen)
Datos (si existen)
20 bytes

- Protocolo de control de transmisión (TCP/IP)
  - Protocolo que permite transmitir grandes bloques de datos sin errores.
  - En Control Winsock oculta la mayor parte de los detalles del protocolo TCP/IP por los que no tenemos que preocuparnos de las banderas ACK ni de los números de secuencia.

El control MSWinsockControl

- Para la creación de aplicaciones basadas en sockets.
- El socket permite a una aplicación establecer una conexión con cualquier punto de la red, así como enviar y recibir datos, principales funciones de cualquier aplicación de Internet.
- Con MSWinsockControl se pueden implementar los diferentes protocolos utilizados en Internet e inclusive protocolos propios

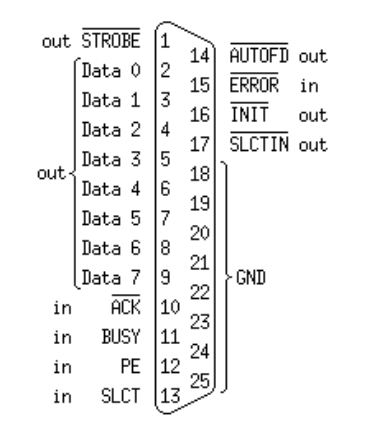
HTTP	Hypertext Transfer Protocol (páginas Web)	puerto 80
FTP	File Transfer Protocol (transferencia de ficheros)	puerto 21
SMTP	Simple Mail Transfer Protocol (envío de email)	puerto 25
POP3	Post Office Protocol (recepción email)	puerto 110
NNTP	Network News Transfer Protocol (noticias)	puerto 119

Bases de Datos

- Registrar en un histórico los diferentes eventos que se producen en las instalaciones para mejorar las tareas de mantenimiento tanto correctivo como preventivo necesario en las instalaciones de Navegación Aérea.
- Base de datos versus fichero de texto.
  - Posibilidad de utilizar toda la potencia que disponen las aplicaciones de gestión de bases de datos para sacar todo el partido a los datos almacenados y de utilizar el estándar ODBC y SQL que permiten utilizar diferentes tipos de bases de datos.
- Tabla histórico formada por tres campos de texto:
  - Nombre, donde se registra el nombre del evento que se ha producido.
  - Estado, donde se registra el estado final del evento que se ha producido.
  - Hora, donde se registra la hora en que se ha producido el evento.

El Puerto Paralelo

- El puerto paralelo de un PC (8 output, 4 input digitales)
  (Out: Pin 2-7, In: Pin 10,12,13,15)



- Podríamos disponer de

$2^8 = 256$ salidas digitales codificadas

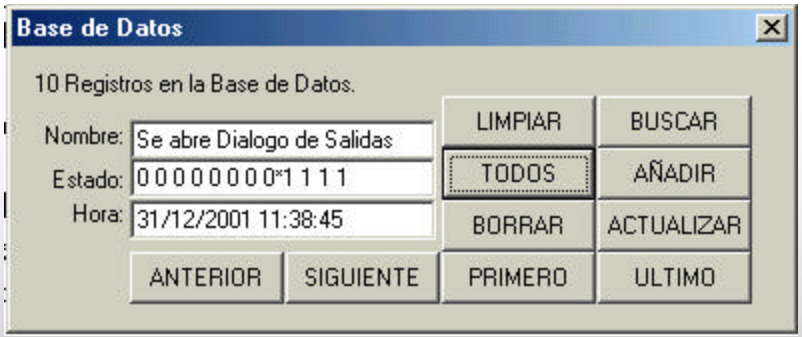
$2^4 = 16$ entradas digitales codificadas

Funcionamiento de la aplicación

- Diálogo Principal

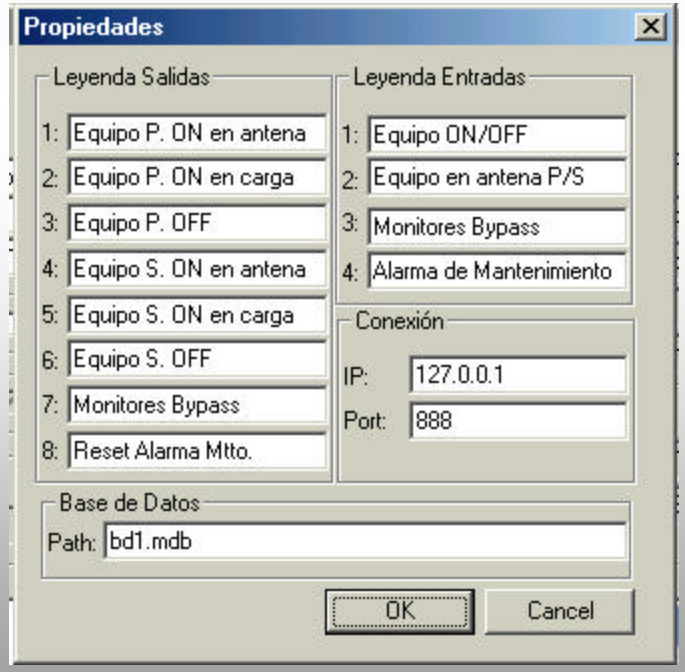
Funcionamiento de la aplicación

- Diálogo Base de datos



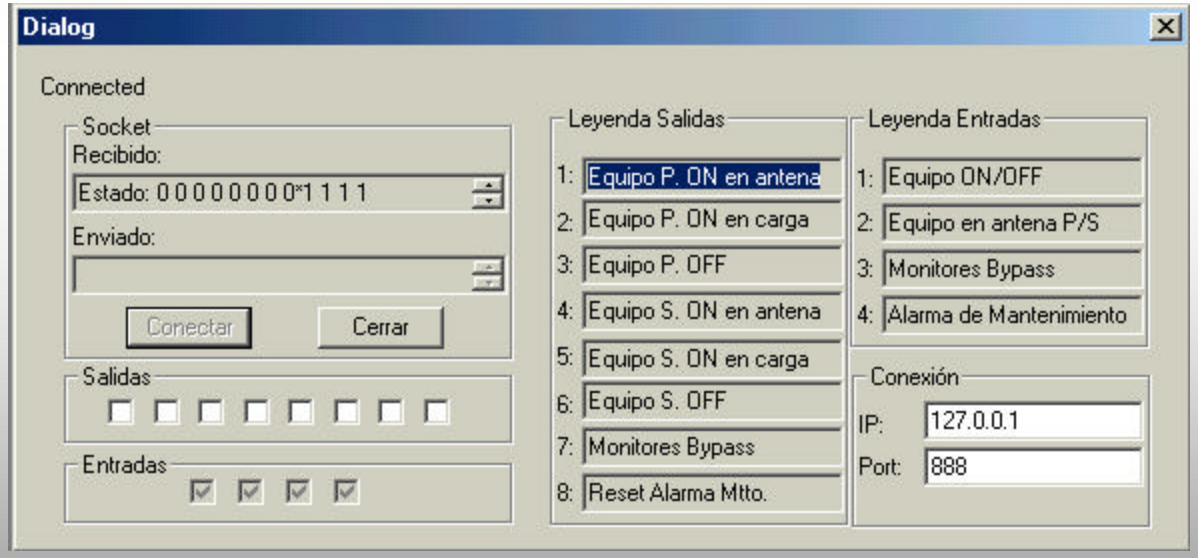
Funcionamiento de la aplicación

- Diálogo Propiedades



Funcionamiento de la aplicación

- Diálogo Cliente



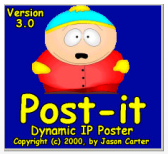
Problema de la IP dinámica

- *Direcciones IP públicas.*
- *Direcciones IP privadas (reservadas).*
- *Direcciones IP estáticas (fijas).*
- *Direcciones IP dinámicas.*

El problema aparece si la dirección IP pública es dinámica, asignada por el proveedor de servicios de Internet, entonces el usuario del ordenador cliente puede no saber cual es la dirección IP del PC que actúa como servidor.

Problema de la IP dinámica

- La solución al problema de la IP dinámica que se ha adoptado es la utilización de un software (Pos_it) que publica en Internet una página web donde puedes saber si el PC servidor esta conectado a la red y cual es su dirección IP publica.



- Una vez ya existe conexión a Internet y ha sido asignada una IP pública dinámica por el proveedor, se debe arrancar este software y entonces se queda residente en el PC, el programa se encarga de publicar cada cierto tiempo una página web con la información requerida actualizada.

Líneas de continuación del Proyecto

- A nivel de comunicación mediante socket's
  - Un socket para comandos y otro socket para datos
  - Varios sockets para conexiones entrantes de varias aplicaciones cliente al mismo servidor y otro para los datos.
  - En visual C++ existe la posibilidad de crear controles en tiempo de ejecución con lo que podríamos crear un socket nuevo con cada conexión entrante.
  - Crear una aplicación servidor con la única funcionalidad de gestionar los mensajes provenientes de las diferentes aplicaciones cliente que se le conectan. Esta posibilidad presenta la ventaja de que se puede instalar la aplicación servidor en una computadora que su dirección IP sea fija y entonces todas las aplicaciones cliente saben donde dirigirse para establecer la comunicación y es la aplicación servidor quien se encargaría de reenviar la información a la aplicación cliente de destino en el caso de estar conectada.

Líneas de continuación del Proyecto

- Crear aplicaciones cliente que enlacen con servidores de POP3 y SMTP (email), http (páginas web), FTP (Transferencia de ficheros),etc. con lo que se puede añadir funciones como alertas por correo electrónico a tu buzón de correo, móvil, etc, presentación de información de la aplicación en una página web, transferencia de las bases de datos en otro ordenador, etc.
- Los socket's pueden ser utilizados con otros lenguajes de programación como ASP que es un lenguaje de programación para realizar páginas web que se ejecutan sobre la maquina servidor. Con esto se puede conseguir que todo el potencial que presentan los socket's se pueda utilizar desde cualquier máquina con cualquier sistema operativo con posibilidad de navegar por Internet o intranet, indiferentemente del navegador que utilice.
- Posibilidad de utilización de password's de acceso a la aplicación para que solo usuarios registrados puedan conectarse y tener acceso a la aplicación
- Posibilidad de encriptar la información que enviamos a la red para evitar posibles usos de información capturada desde la red

Líneas de continuación del Proyecto

- A nivel de funcionalidad de la aplicación servidor en nuestra aplicación se ha optado por el control del puerto paralelo con lo que ya se obtienen una amplia gama de funciones en muchos campos.
  - Se podría añadir el control de alguna cámara de video que publique las imágenes en Internet. A través de nuestra aplicación podríamos controlar la dirección, selección, zoom, etc.
  - Añadir comunicación mediante RS232 de nuestra aplicación con un microcontrolador externo o un PLC. Con esto podríamos aprovechar todas las posibilidades del microcontrolador elegido y controlarlas desde una aplicación en un PC remoto. A partir de aquí se puede realizar una red por cable o inalámbrica de microcontroladores esclavos con un microcontrolador master conectado al PC.
  - Una opción muy interesante sería la de poder controlar el ordenador servidor telefónicamente para poder arrancarlo, conectarse a la red o arrancar la aplicación cliente-servidor que se desee.

Líneas de continuación del Proyecto

- Crear aplicaciones cliente-servidor mediante microcontroladores y así no utilizar un PC que es mucho más caro. Las aplicaciones servidor se implementan en un chip (micro servidor) que dispone de una pila TCP/IP y de un modem o una adaptador de red para conectarse a la red. El circuito integrado estará conectado a la red y podrá utilizar socket's, emails, páginas web, etc y a partir de aquí conectarse con PC o con otros microservidores que realizan las funciones necesarias en cada caso. Con esto podríamos añadir a cualquier electrodoméstico, a cualquier electroválvula, a cualquier sensor funcionalidad cliente-servidor y ser controlado desde cualquier punto de la red en los que se conecten.

Demostración

ProyectoBD.exe